

Exhibit H

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.709/Y.1331

(06/2016)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,
DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – General

SERIES Y: GLOBAL INFORMATION
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS
AND NEXT-GENERATION NETWORKS, INTERNET OF
THINGS AND SMART CITIES

Internet protocol aspects – Transport

Interfaces for the optical transport network

Recommendation ITU-T G.709/Y.1331

ITU-T G-SERIES RECOMMENDATIONS
TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
General	G.700–G.709
Coding of voice and audio signals	G.710–G.729
Principal characteristics of primary multiplex equipment	G.730–G.739
Principal characteristics of second order multiplex equipment	G.740–G.749
Principal characteristics of higher order multiplex equipment	G.750–G.759
Principal characteristics of transcoder and digital multiplication equipment	G.760–G.769
Operations, administration and maintenance features of transmission equipment	G.770–G.779
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.780–G.789
Other terminal equipment	G.790–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.709/Y.1331

Interfaces for the optical transport network

Summary

Recommendation ITU-T G.709/Y.1331 defines the requirements for the optical transport network (OTN) interface signals of the optical transport network, in terms of:

- OTN hierarchy
- functionality of the overhead in support of multi-wavelength optical networks
- frame structures
- bit rates
- formats for mapping client signals.

Edition 5.0 of this Recommendation includes the text of Amendments 2, 3, 4, Corrigendum 2 and Erratum 1 to Edition 4.0 of this Recommendation, support for frequency and time synchronisation, addition of Beyond 100 Gbit/s OTU, ODU and OPU frame formats, overhead, ODU multiplexing, client mappings and optical layer terminology updates. Edition 5.0 furthermore deleted the OPUk concatenation specifications and the ATM mapping into OPUk specification and changed the TCM ACT and FTFL overhead bytes into EXP overhead bytes.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.709/Y.1331	2001-02-09	15	11.1002/1000/5350
1.1	ITU-T G.709/Y.1331 (2001) Amd. 1	2001-11-29	15	11.1002/1000/5629
2.0	ITU-T G.709/Y.1331	2003-03-16	15	11.1002/1000/6265
2.1	ITU-T G.709/Y.1331 (2003) Amd. 1	2003-12-14	15	11.1002/1000/7060
2.2	ITU-T G.709/Y.1331 (2003) Cor. 1	2006-12-14	15	11.1002/1000/8982
2.3	ITU-T G.709/Y.1331 (2003) Amd. 2	2007-11-22	15	11.1002/1000/9155
2.4	ITU-T G.709/Y.1331 (2003) Cor.2	2009-01-13	15	11.1002/1000/9646
2.5	ITU-T G.709/Y.1331 (2003) Amd. 3	2009-04-22	15	11.1002/1000/9671
3.0	ITU-T G.709/Y.1331	2009-12-22	15	11.1002/1000/10398
3.1	ITU-T G.709/Y.1331 (2009) Cor. 1	2010-07-29	15	11.1002/1000/10875
3.2	ITU-T G.709/Y.1331 (2009) Amd. 1	2010-07-29	15	11.1002/1000/10874
3.3	ITU-T G.709/Y.1331 (2009) Amd. 2	2011-04-13	15	11.1002/1000/11115
4.0	ITU-T G.709/Y.1331	2012-02-13	15	11.1002/1000/11485
4.1	ITU-T G.709/Y.1331 (2012) Cor. 1	2012-10-29	15	11.1002/1000/11776
4.2	ITU-T G.709/Y.1331 (2012) Amd. 1	2012-10-29	15	11.1002/1000/11774
4.3	ITU-T G.709/Y.1331 (2012) Amd. 2	2013-10-22	15	11.1002/1000/11982
4.4	ITU-T G.709/Y.1331 (2012) Amd. 3	2014-12-05	15	11.1002/1000/12363
4.5	ITU-T G.709/Y.1331 (2012) Cor. 2	2015-01-13	15	11.1002/1000/12365
4.6	ITU-T G.709/Y.1331 (2012) Amd. 4	2015-01-13	15	11.1002/1000/12364
5.0	ITU-T G.709/Y.1331	2016-06-22	15	11.1002/1000/12789

Keywords

Client mappings, frame formats, multiplexing, OTN.

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2016

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	3
3.1 Terms defined elsewhere	3
3.2 Terms defined in this Recommendation.....	4
4 Abbreviations and acronyms	4
5 Conventions	9
6 Optical transport network interface structure	9
6.1 Basic signal structure.....	10
6.2 Information structure for OTN interfaces.....	12
7 Multiplexing/mapping principles and bit rates	16
7.1 Mapping.....	18
7.2 Wavelength division multiplex.....	18
7.3 Bit rates and capacity.....	18
7.4 ODUk time-division multiplex.....	23
8 OTN Interfaces	30
8.1 Single-OTU (SOTU) interface	30
8.2 Multi-OTU (MOTU) interface	30
8.3 Single-OTU with management (SOTUm) interface.....	31
8.4 Multi-OTU with management (MOTUm) interface.....	31
9 Media Element.....	31
10 OCh and OTSiA	31
10.1 OCh.....	31
10.2 Optical tributary signal assembly (OTSiA).....	31
11 Optical transport unit (OTU)	32
11.1 OTUk frame structure.....	32
11.2 Scrambling.....	33
11.3 OTUCn frame structure	33
12 Optical data unit (ODU)	34
12.1 ODU frame structure	34
12.2 ODU bit rates and bit-rate tolerances	35
13 Optical payload unit (OPU).....	38
14 Overhead information carried over the OSC and OCC	38
15 Overhead description	39
15.1 Types of overhead	43
15.2 Trail trace identifier and access point identifier definition	44
15.3 OTS-O description.....	46

	Page
15.4 OMS-O description	47
15.5 OCh-O and OTSiG-O description	47
15.6 OTU/ODU frame alignment OH description	49
15.7 OTU OH description	50
15.8 ODU OH description	56
15.9 OPU OH description	70
16 Maintenance signals	72
16.1 OTS maintenance signals	73
16.2 OMS maintenance signals	73
16.3 OCh and OTiSA maintenance signals	73
16.4 OTU maintenance signals	74
16.5 ODU maintenance signals	75
16.6 Client maintenance signal	77
17 Mapping of client signals	77
17.1 OPU client signal fail (CSF)	78
17.2 Mapping of CBR2G5, CBR10G, CBR10G3 and CBR40G signals into OPUk	78
17.3 Blank clause	82
17.4 Mapping of GFP frames into OPUk (k=0,1,2,3,4,flex)	82
17.5 Mapping of test signal into OPU	83
17.6 Mapping of a non-specific client bit stream into OPUk	84
17.7 Mapping of other constant bit-rate signals with justification into OPUk	85
17.8 Mapping a 1000BASE-X and FC-1200 signal via timing transparent transcoding into OPUk	95
17.9 Mapping a supra-2.488 Gbit/s CBR signal into OPUflex using BMP	97
17.10 Mapping of packet client signals into OPUk	100
17.11 Mapping of FlexE client signals into OPUflex using IMP	100
17.12 Mapping of FlexE aware signals into OPUflex	101
18 Blank clause	105
19 Mapping ODU _j signals into the ODTU signal and the ODTU into the OPUk tributary slots	105
19.1 OPUk tributary slot definition	105
19.2 ODTU definition	114
19.3 Multiplexing ODTU signals into the OPUk	116
19.4 OPUk multiplex overhead and ODTU justification overhead	124
19.5 Mapping ODU _j into ODTU _{jk}	135
19.6 Mapping of ODU _j into ODTU _k .ts	143
20 Mapping ODU _k signals into the ODTUC _n signal and the ODTUC _n into the OPUC _n tributary slots	147
20.1 OPUC _n tributary slot definition	147

	Page
20.2 ODTUCn definition.....	151
20.3 Multiplexing ODTUCn signals into the OPUCn.....	152
20.4 OPUCn multiplex overhead and OTU justification overhead.....	153
20.5 Mapping ODUk into ODTUCn.ts	157
Annex A – Forward error correction using 16-byte interleaved RS(255,239) codecs	160
Annex B – Adapting 64B/66B encoded clients via transcoding into 513B code blocks.....	162
B.1 Transmission order	162
B.2 Client frame recovery	162
B.3 Transcoding from 66B blocks to 513B blocks	162
B.4 Link fault signalling	167
Annex C – Adaptation of OTU3 and OTU4 over multichannel parallel interfaces	168
Annex D – Generic mapping procedure principles.....	171
D.1 Basic principle	171
D.2 Applying GMP in OTN	174
D.3 $C_m(t)$ encoding and decoding	178
D.4 $\Sigma C_{nD}(t)$ encoding and decoding.....	184
Annex E – Adaptation of parallel 64B/66B encoded clients	186
E.1 Introduction	186
E.2 Clients signal format.....	186
E.3 Client frame recovery	186
E.4 Additions to Annex B transcoding for parallel 64B/66B clients.....	189
Annex F – Improved robustness for mapping of 40GBASE-R into OPU3 using 1027B code blocks	192
F.1 Introduction	192
F.2 513B code block framing and flag bit protection.....	192
F.3 66B block sequence check.....	193
Annex G – Mapping ODU0 into a low latency OTU0 (OTU0LL)	197
G.1 Introduction	197
G.2 Optical transport unit 0 low latency (OTU0LL).....	197
Annex H – OTUCn sub rates (OTUCn-M).....	200
H.1 Introduction	200
H.2 OTUCn-M frame format	200
Appendix I – Range of stuff ratios for asynchronous mappings of CBR2G5, CBR10G, and CBR40G clients with ± 20 ppm bit-rate tolerance into OPUK, and for asynchronous multiplexing of ODUj into ODUk ($k > j$).....	201
Appendix II – Examples of functionally standardized OTU frame structures	207
Appendix III – Example of ODUk multiplexing	210
Appendix IV – Blank appendix	212
Appendix V – ODUk multiplex structure identifier (MSI) examples	213

	Page
Appendix VI – Parallel logic implementation of the CRC-9, CRC-8, CRC-5 and CRC-6	215
Appendix VII – OTL4.10 structure.....	218
Appendix VIII – CPRI into ODU mapping	219
Appendix IX – Overview of CBR clients into OPU mapping types	220
Appendix X – Overview of ODU _j into OPU _k mapping types	222
Appendix XI – Derivation of recommended ODUflex(GFP) bit-rates and examples of ODUflex(GFP) clock generation.....	224
XI.1 Introduction	224
XI.2 Tributary slot sizes	224
XI.3 Example methods for ODUflex(GFP) clock generation	227
Appendix XII – Terminology changes between ITU-T G.709 Edition 4 and Edition 5.....	229
Appendix XIII – OTUC _n sub rates (OTUC _n -M) Applications	231
XIII.1 Introduction	231
XIII.2 OTUC _n -M frame format and rates	231
XIII.3 OTUC _n -M fault condition.....	232
Bibliography.....	233

Recommendation ITU-T G.709/Y.1331

Interfaces for the optical transport network

1 Scope

The optical transport network (OTN) supports the operation and management aspects of optical networks of various architectures, e.g., point-to-point, ring and mesh architectures.

This Recommendation defines the interfaces of the OTN to be used within and between subnetworks of the optical network, in terms of:

- OTN hierarchy;
- functionality of the overhead in support of multi-wavelength optical networks;
- frame structures;
- bit rates;
- formats for mapping client signals.

The interfaces defined in this Recommendation can be applied at user-to-network interfaces (UNI) and network node interfaces (NNI) of the OTN.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- | | |
|-----------------|---|
| [ITU-T G.652] | Recommendation ITU-T G.652 (2009), <i>Characteristics of a single-mode optical fibre and cable</i> . |
| [ITU-T G.693] | Recommendation ITU-T G.693 (2009), <i>Optical interfaces for intra-office systems</i> . |
| [ITU-T G.695] | Recommendation ITU-T G.695 (2015), <i>Optical interfaces for coarse wavelength division multiplexing applications</i> . |
| [ITU-T G.698.1] | Recommendation ITU-T G.698.1 (2009), <i>Multichannel DWDM applications with single-channel optical interfaces</i> . |
| [ITU-T G.698.2] | Recommendation ITU-T G.698.2 (2009), <i>Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces</i> . |
| [ITU-T G.698.3] | Recommendation ITU-T G.698.3 (2012), <i>Multichannel seeded DWDM applications with single-channel optical interfaces</i> . |
| [ITU-T G.707] | Recommendation ITU-T G.707/Y.1322 (2007), <i>Network node interface for the synchronous digital hierarchy (SDH)</i> . |
| [ITU-T G.780] | Recommendation ITU-T G.780/Y.1351 (2010), <i>Terms and definitions for synchronous digital hierarchy (SDH) networks</i> . |
| [ITU-T G.798] | Recommendation ITU-T G.798 (2012), <i>Characteristics of optical transport network hierarchy equipment functional blocks</i> . |

- [ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.
- [ITU-T G.806] Recommendation ITU-T G.806 (2012), *Characteristics of transport equipment – Description methodology and generic functionality*.
- [ITU-T G.870] Recommendation ITU-T G.870/Y.1352 (2016), *Terms and definitions for optical transport networks (OTN)*.
- [ITU-T G.872] Recommendation ITU-T G.872 (2012), *Architecture of optical transport networks*.
- [ITU-T G.873.1] Recommendation ITU-T G.873.1 (2014), *Optical Transport Network (OTN): Linear protection*.
- [ITU-T G.873.2] Recommendation ITU-T G.873.2 (2015), *ODUk shared ring protection (SRP)*.
- [ITU-T G.959.1] Recommendation ITU-T G.959.1 (2016), *Optical transport network physical layer interfaces*.
- [ITU-T G.984.6] Recommendation ITU-T G.984.6 (2008), *Gigabit-capable passive optical networks (GPON): Reach extension*.
- [ITU-T G.987.4] Recommendation ITU-T G.987.4 (2012), *10 Gigabit-capable passive optical networks (XG-PON): Reach extension*.
- [ITU-T G.7041] Recommendation ITU-T G.7041/Y.1303 (2016), *Generic framing procedure (GFP)*.
- [ITU-T G.7044] Recommendation ITU-T G.7044/Y.1347 (2011), *Hitless Adjustment of ODUflex(GFP)*.
- [ITU-T G.7712] Recommendation ITU-T G.7712/Y.1703 (2010), *Architecture and specification of data communication network*.
- [ITU-T G.7714.1] Recommendation ITU-T G.7714.1/Y.1705.1 (2010), *Protocol for automatic discovery in SDH and OTN networks*.
- [ITU-T G.8011.1] Recommendation ITU-T G.8011.1/Y.1307.1 (2009), *Ethernet private line service*.
- [ITU-T G.8260] Recommendation ITU-T G.8260 (2015), *Definitions and terminology for synchronization in packet networks*.
- [ITU-T G.8601] Recommendation ITU-T G.8601/Y.1391 (2006), *Architecture of service management in multi-bearer, multi-carrier environment*.
- [ITU-T M.1400] Recommendation ITU-T M.1400 (2001), *Designations for interconnections among operators' networks*.
- [ITU-T M.3100 Amd.3] Recommendation ITU-T M.3100 (1995) Amd.3 (2001), *Generic network information model – Amendment 3: Definition of the management interface for a generic alarm reporting control (ARC) feature*.
- [ITU-T O.150] Recommendation ITU-T O.150 (1996), *General requirements for instrumentation for performance measurements on digital transmission equipment*.
- [ITU-T T.50] Recommendation ITU-T T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.

- [IEEE 802.3] IEEE Std. 802.3:2015, *IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*.
- [OIF FlexE IA] OIF, *Flex Ethernet Implementation Agreement* (2016).

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 Terms defined in [ITU-T G.780]:

- BIP-X
- network node interface

3.1.2 Terms defined in [ITU-T G.805]:

- adapted information (AI)
- characteristic information (CI)
- network
- subnetwork

3.1.3 Terms defined in [ITU-T G.870]:

- CBR10G
- CBR2G5
- CBR40G
- completely standardized OTUk (OTUk)
- connection monitoring end point (CMEP)
- functionally standardized OTUk (OTUkV)
- hitless activation/deactivation of a connection monitor
- non associated overhead (naOH)
- ODUk path (ODUkP)
- ODUk TCM (ODUkT)
- optical data unit (ODUk)
- optical payload unit (OPUk)
- optical transport unit (OTUk[V])
-
- optical physical section (OPS)
- optical supervisory channel (OSC)
- optical transport network (OTN)
- optical transport network node interface (ONNI)

3.1.4 Terms defined in [ITU-T G.872]:

- optical multiplex section (OMS)
- optical transmission section (OTS)

3.1.5 Terms defined in [ITU-T G.959.1]:

- optical tributary signal (OTSi).

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 ODUk.ts: The ODUk.ts is an increment of bandwidth which when multiplied by a number of tributary slots gives the recommended size of an ODUFlex(GFP) optimized to occupy a given number of tributary slots of an OPUk.

3.2.2 OTU0 low latency: The OTU0 low latency (OTU0LL) is the information structure used for transport of an ODU0 over a multi-vendor optical network interface in one administrative domain at the edge of the optical transport network.

NOTE – The OTU0LL is not transported over an SOTU, MOTUm, MOTU.

3.2.3 optical tributary signal assembly (OTSiA): The OTSiG together with the non-associated overhead (OTSiG-O).

3.2.4 optical tributary signal group (OTSiG): The set of OTSi signals that supports an OTU.

3.2.5 optical tributary signal overhead (OTSiG-O): The non-associated overhead for an OTSiG.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

16FS	16 columns with Fixed Stuff
3R	Reamplification, Reshaping and Retiming
AI	Adapted Information
AIS	Alarm Indication Signal
AMP	Asynchronous Mapping Procedure
API	Access Point Identifier
APS	Automatic Protection Switching
ASI	Asynchronous Serial Interface for DVB
BDI	Backward Defect Indication
BDI-O	Backward Defect Indication Overhead
BDI-P	Backward Defect Indication Payload
BEI	Backward Error Indication
BI	Backward Indication
BIAE	Backward Incoming Alignment Error
BIP	Bit Interleaved Parity
BMP	Bit-synchronous Mapping Procedure
CAUI	(Chip to) 100 Gb/s Attachment Unit Interface
CB	Control Block
CBR	Constant Bit Rate
CI	Characteristic Information

CM	Connection Monitoring
C_m	number of m-bit Client data entities
CMEP	Connection Monitoring End Point
CMGPON_D	Continuous Mode GPON Downstream
CMGPON_U2	Continuous Mode GPON Upstream 2
CMOH	Connection Monitoring Overhead
CMXGPON_D	Continuous Mode XGPON Downstream
CMXGPON_U2	Continuous Mode XGPON Upstream 2
C_n	number of n-bit client data entities
C_{nD}	difference between C_n and $(m/n \times C_m)$
CPRI	Common Public Radio Interface
CRC	Cyclic Redundancy Check
CS	Client Specific
CSF	Client Signal Fail
CTRL	Control word sent from source to sink
DAPI	Destination Access Point Identifier
DDR	Double Data Rate
DMp	Delay Measurement of ODUk path
DMti	Delay Measurement of TCMi
DNU	Do Not Use
DVB	Digital Video Broadcast
EDC	Error Detection Code
EOS	End Of Sequence
ESCON	Enterprise Systems Connection
EXP	Experimental
ExTI	Expected Trace Identifier
FA	Frame Alignment
FAS	Frame Alignment Signal
FC	Fibre Channel
FC	Flag Continuation
FDI	Forward Defect Indication
FDI-O	Forward Defect Indication Overhead
FDI-P	Forward Defect Indication Payload
FEC	Forward Error Correction
GCC	General Communication Channel
GID	Group Identification
GMP	Generic Mapping Procedure

GPON	Gigabit-capable Passive Optical Networks
IAE	Incoming Alignment Error
IB	InfiniBand
IMP	Idle Mapping Procedure
IP	Internet Protocol
JC	Justification Control
JOH	Justification Overhead
LF	Local Fault
LLM	Logical Lane Marker
LSB	Least Significant Bit
MFAS	MultiFrame Alignment Signal
MFI	Multiframe Indicator
MOTU	Multi-OTU
MOTUm	Multi-OTU with management
MPLS	Multi Protocol Label Switching
MPLS-TP	Multi Protocol Label Switching – Transport Profile
MS	Maintenance Signal
MSB	Most Significant Bit
MSI	Multiplex Structure Identifier
naOH	non-associated Overhead
NJO	Negative Justification Opportunity
NNI	Network Node Interface
NOS	Not_Operational Sequence
OCC	Overhead Communication Channel
OCI	Open Connection Indication
ODTUG	Optical Data Tributary Unit Group
ODTUGk/Cn	Optical Data Tributary Unit Group-k/Cn
ODTUjk	Optical Data Tributary Unit j into k
ODTUK/Cn.ts	Optical Data Tributary Unit k/Cn with ts tributary slots
ODU	Optical Data Unit
ODUk/Cn	Optical Data Unit-k/Cn
ODUk/Cn.ts	Optical Data Unit k/Cn fitting in ts tributary slots
ODUk/CnP	Optical Data Unit-k/Cn Path monitoring level
ODUk/CnT	Optical Data Unit-k/Cn Tandem connection monitoring level
OH	Overhead
OMFI	OPU Multi-Frame Identifier
OMS	Optical Multiplex Section

OMS-O	Optical Multiplex Section Overhead
ONNI	Optical Network Node Interface
OPS	Optical Physical Section
OPSM	Optical Physical Section Multilane
OPU	Optical Payload Unit
OPUk/Cn	Optical Payload Unit-k/Cn
OSC	Optical Supervisory Channel
OSMC	OTN synchronization messaging channel
OTL	Optical Transport Lane
OTLk.n	group of n Optical Transport Lanes that carry one OTUk
OTLC.n	group of n Optical Transport Lanes that carry one OTUC of an OTUCn
OTN	Optical Transport Network
OTS	Optical Transmission Section
OTS-O	Optical Transmission Section Overhead
OTSi	Optical Tributary Signal
OTSiA	Optical Tributary Signal Assembly
OTSiG	Optical Tributary Signal Group
OTSiG-O	Optical Tributary Signal Group - Overhead
OTU	Optical Transport Unit
OTU0LL	Completely standardized Optical Transport Unit-0 Low Latency
OTUCn-M	Optical Transport Unit-Cn with n OxUC overhead instances and M 5G tributary slots
OTUk/Cn	completely standardized Optical Transport Unit-k/Cn
OTUkV	functionally standardized Optical Transport Unit-k
OTUk-v	Optical Transport Unit-k with vendor specific OTU FEC
PCC	Protection Communication Channel
P-CMEP	Path-Connection Monitoring End Point
PCS	Physical Coding Sublayer
PJO	Positive Justification Opportunity
PKT	Packet
PLD	Payload
PM	Path Monitoring
PMA	Physical Medium Attachment sublayer
PMI	Payload Missing Indication
PMOH	Path Monitoring Overhead
PN	Pseudo-random Number
POS	Position field

ppm	parts per million
PRBS	Pseudo Random Binary Sequence
PSI	Payload Structure Identifier
PT	Payload Type
QDR	Quad Data Rate
RES	Reserved for future international standardization
RF	Remote Fault
RS	Reed-Solomon
RS-Ack	Re-sequence Acknowledge
SAPI	Source Access Point Identifier
SBCON	Single-Byte command code sets Connection
SDI	Serial Digital Interface
SDR	Single Data Rate
Sk	Sink
SM	Section Monitoring
SMOH	Section Monitoring Overhead
SNC	Subnetwork Connection
SNC/I	Subnetwork Connection protection with Inherent monitoring
SNC/N	Subnetwork Connection protection with Non-intrusive monitoring
SNC/S	Subnetwork Connection protection with Sublayer monitoring
So	Source
SOTU	Single-OTU
SOTUm	Single-OTU with management
TC	Tandem Connection
TC-CMEP	Tandem Connection-Connection Monitoring End Point
TCM	Tandem Connection Monitoring
TCMOH	Tandem Connection Monitoring Overhead
TS	Tributary Slot
TSI	Transmitter Structure Identifier
TSOH	Tributary Slot Overhead
TTI	Trail Trace Identifier
TTT	Timing Transparent Transcoding
TxTI	Transmitted Trace Identifier
UNI	User-to-Network Interface
XC	Cross Connect
XGPON	10 Gigabit-capable Passive Optical Networks

5 Conventions

This Recommendation uses the following conventions defined in [ITU-T G.870]:

- k
- m
- n

The functional architecture of the optical transport network as specified in [ITU-T G.872] is used to derive the ONNI. The ONNI is specified in terms of the adapted and characteristic information present in each layer as described in [ITU-T G.805].

Transmission order: The order of transmission of information in all the diagrams in this Recommendation is first from left to right and then from top to bottom. Within each byte the most significant bit is transmitted first. The most significant bit (bit 1) is illustrated at the left in all the diagrams.

Mapping order: The serial bit stream of a constant bit rate signal is inserted into the OPU payload so that the bits will be transmitted on the OPU/ODU in the same order that they were received at the input of the AMP, BMP or GMP mapper function. If m bits b_a, b_b, b_c up to b_m are client signal bits of which b_a is the bit that is received first and b_m is the bit that is received last, then b_a will be mapped into bit 1 of a first OPU byte and b_m will be mapped into bit 8 of an n^{th} OPU byte (with $n = m/8$).

Value of reserved bit(s): The value of an overhead bit, which is reserved or reserved for future international standardization shall be set to "0".

Value of non-sourced bit(s): Unless stated otherwise, any non-sourced bits shall be set to "0".

OTUk, OTUCn, ODUk, ODUCn, OPUk and OPUCn overhead assignment: The assignment of an overhead in the optical transport/data/payload unit signal to each part is defined in Figure 5-1. OTUk, ODUk, OPUk contain one instance of OTU, ODU, OPU overhead. OTUCn, ODUCn and OPUCn contain n instances of OTU, ODU, OPU overhead, numbered 1 to n .

Interleaved versions of the OTU, ODU and OPU overhead may be present on OTUCn interfaces. This interleaving is interface specific and specified for OTN interfaces with standardized application codes in the interface specific Recommendations (ITU-T G.709.x series). Within the other clauses of this Recommendation an OTUCn, ODUCn and OPUCn are presented in an interface independent manner, by means of n OTUC, ODUC and OPUC instances that are marked #1 to # n .

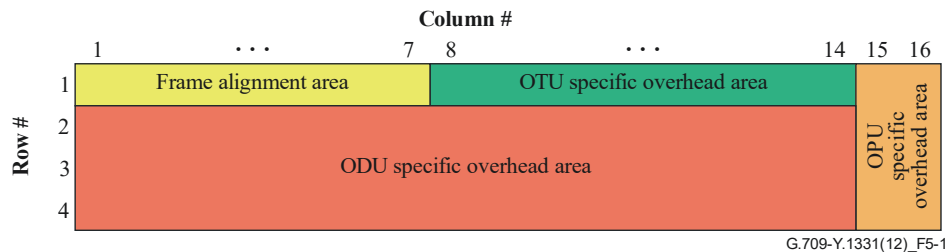


Figure 5-1 – OTU, ODU and OPU overhead

6 Optical transport network interface structure

The optical transport network as specified in [ITU-T G.872] defines two interface types, each with two sub-types:

- Single optical transport unit interfaces:
 - without optical layer overhead (SOTU);
 - with optical layer overhead (SOTUm).

- Multi optical transport unit interfaces:
 - without optical layer overhead (MOTU);
 - with optical layer overhead (MOTUm).

The OTN interfaces without optical layer overhead are defined with 3R processing at each end of the interface.

The interfaces consist of digital and management information defined in this Recommendation as carried by a physical interface, the specifications of which are outside the scope of this Recommendation.

Each interface may be associated with one or more application identifiers. Each application identifier represents either a standardized application code or a vendor specific identifier. An interface with an application code may be used to interconnect (OTN) equipment from different vendors. An interface with a vendor specific identifier is to be used to interconnect (OTN) equipment from the same vendor.

6.1 Basic signal structure

The basic structure is shown in Figures 6-1 and 6-2 and consists of a digital and an optical structure.

6.1.1 OTN digital structure

The OTN digital structure (see Figure 6-1) consists of two classes of optical transport units (OTU); the OTU_k and OTU_{Cn}. The OTU_k signal consists of a 4080 column by 4 row frame, which includes 256 columns allocated to contain a forward error correction code (FEC) and is operated at various bit rates that are represented by the value of k. The OTU_{Cn} signal consists of n interleaved 3824 column by 4 row frames, which do not include a FEC area and is operated at n times a basic rate that is represented by OTU_C. FEC for the OTU_{Cn} signal is interface specific and not included in the OTU_{Cn} definition.

The OTU_k contains an optical data unit (ODU_k) and the ODU_k contains an optical payload unit (OPU_k). The OTU_k and its ODU_k perform digital section and path layer roles.

The OTU_{Cn} contains an optical data unit (ODU_{Cn}) and the ODU_{Cn} contains an optical payload unit (OPU_{Cn}). The OTU_{Cn} and its ODU_{Cn} perform digital section layer roles only.

- The completely standardized OTU_k or functionally standardized OTU_{k-v} and OTU_{kV} provide supervision and conditions the signal for transport between 3R regeneration points in the OTN.
- The ODU_k which provides:
 - tandem connection monitoring (ODU_{kT})
 - end-to-end path supervision (ODU_{kP})
 - adaptation of client signals via the OPU_k
 - adaptation of client ODU_k signals via the OPU_k.
- The OTU_{Cn} provides supervision for transport between 3R regeneration points in the OTN. Note that the OTU_{Cn} does not condition the signal for transport between 3R regeneration points because encapsulation and forward error correction are not included in the OTU_{Cn} definition.
- The ODU_{Cn} provides supervision for transport between ODU_k aware points in the OTN.
 - tandem connection monitoring (ODU_{CnT})
 - end-to-end section supervision (ODU_{CnP})
 - adaptation of ODU_k signals via the OPU_{Cn}.

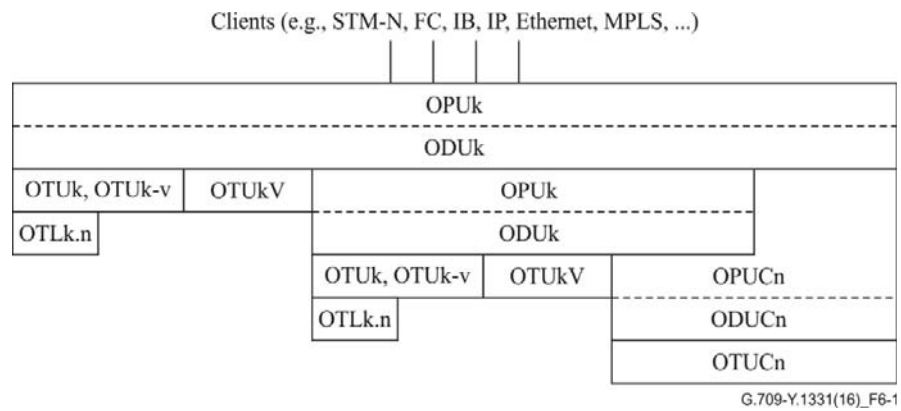


Figure 6-1 – Digital structure of the OTN interfaces

6.1.2 OTN optical structure

The OTN optical structure (see Figure 6-2) consists of two classes of optical interfaces: the single optical transport unit interfaces (SOTU, SOTUm) and the multi optical transport unit interfaces (MOTU, MOTUm).

These interfaces may support optical layer overhead (represented by the *-O entities in Figure 6-2, top). Such overhead may be transported within the optical supervisory channel (OSC), the overhead communication channel (OCC) that is provided by the overhead communication network (OCN) (refer to [ITU-T G.7712]) or an alternative communication channel. Interfaces that support OCh-O and/or OTSiG-O support switching in the optical layer of the OCh and/or OTSiA signals which carry one optical transport unit signal between 3R regeneration points. Interfaces that support OTS-O and OMS-O also support deployment of in-line optical amplifiers between optical layer switching points.

Interfaces that do not support optical layer overhead are designed to transport the optical transport unit signals over a single optical span with 3R regeneration points at each end. For such cases there are no OCh or OTSiA layers present in the interface stack (see Figure 6-2, bottom).

NOTE – Figure 6-2 (bottom) describes the optical structure of OTUk interfaces. The optical structure of OTUCn interfaces with standardized application code are described in the interface specific Recommendations (ITU-T G.709.x series).

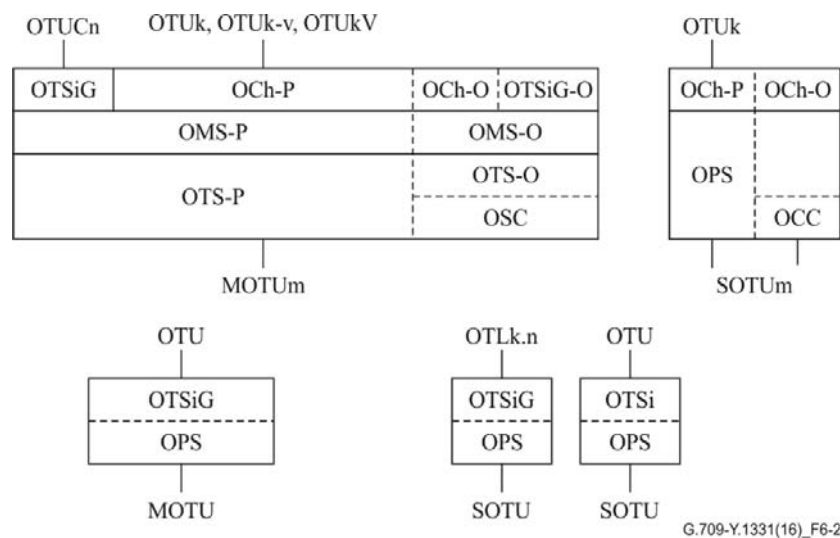
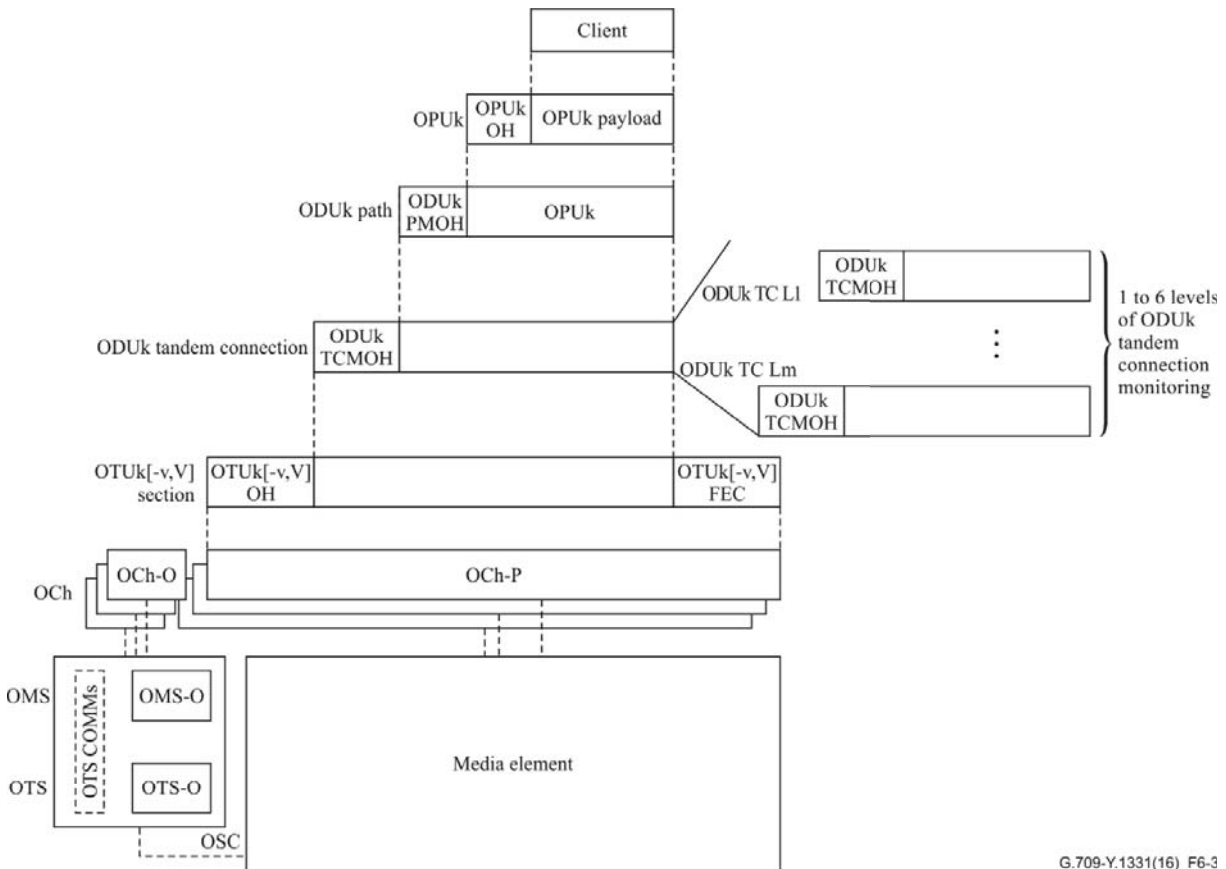


Figure 6-2 – Optical structure of the OTN interfaces

6.2 Information structure for OTN interfaces

The information structure for OTN interfaces is represented by information containment relationships and flows. The principal information containment relationships are described in Figures 6-3, 6-4, 6-5, 6-6, 6-7 and 6-8.

For supervision purposes in OTN interfaces with optical layer overhead, the OTUk/OTUk-v/OTUkV signal is terminated whenever the OCh signal is terminated and the OTUCn signal may be terminated when the OTSiA signal is terminated. For supervision purposes in OTN interfaces without optical layer overhead, the OTUk/OTUk-v/OTUkV and OTUCn signals are terminated whenever the interface signal is terminated.



G.709-Y.1331(16)_F6-3

Figure 6-3 – MOTUm interface principal information containment relationships

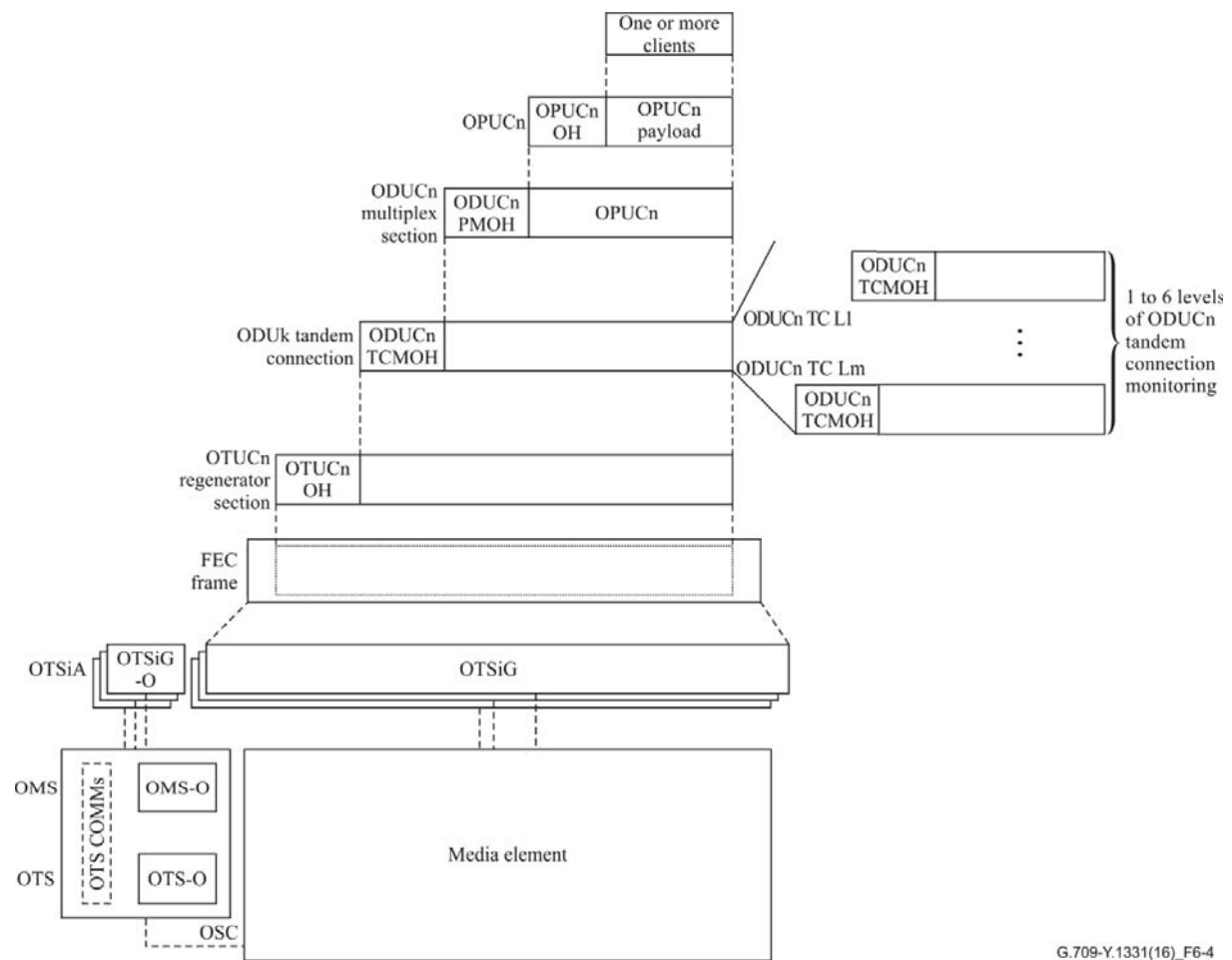


Figure 6-4 – B100G MOTUm interface principal information containment relationships

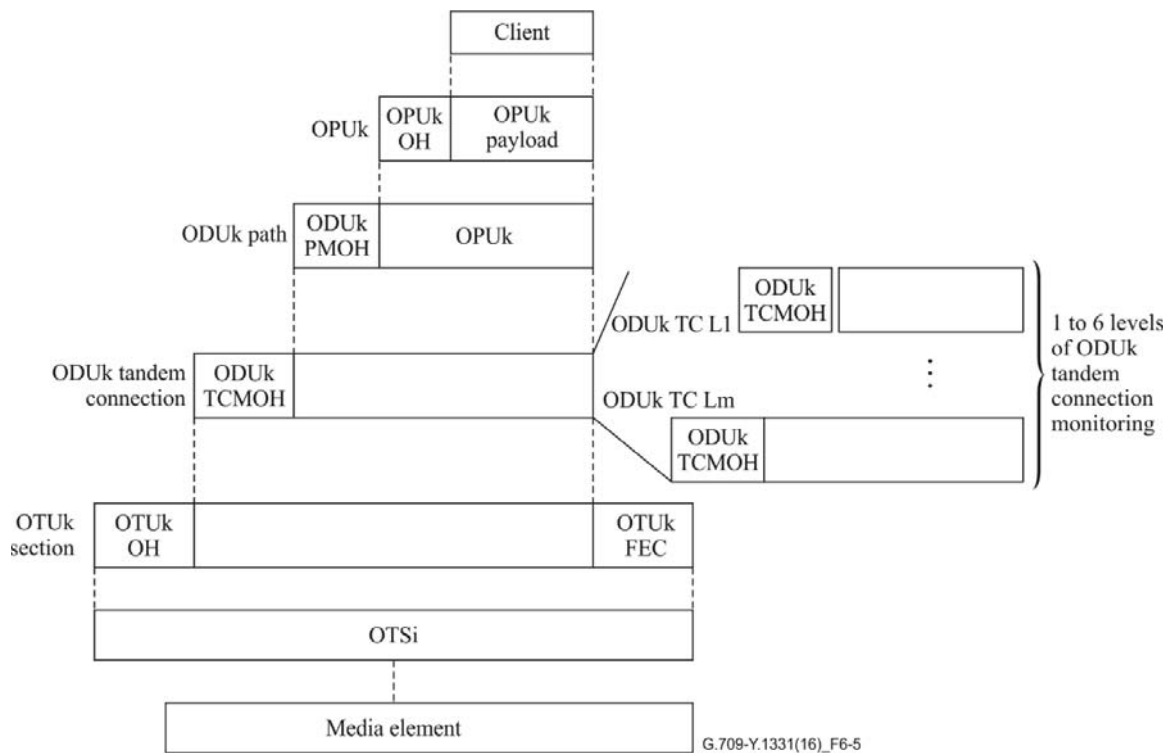


Figure 6-5 – SOTU interface principal information containment relationships

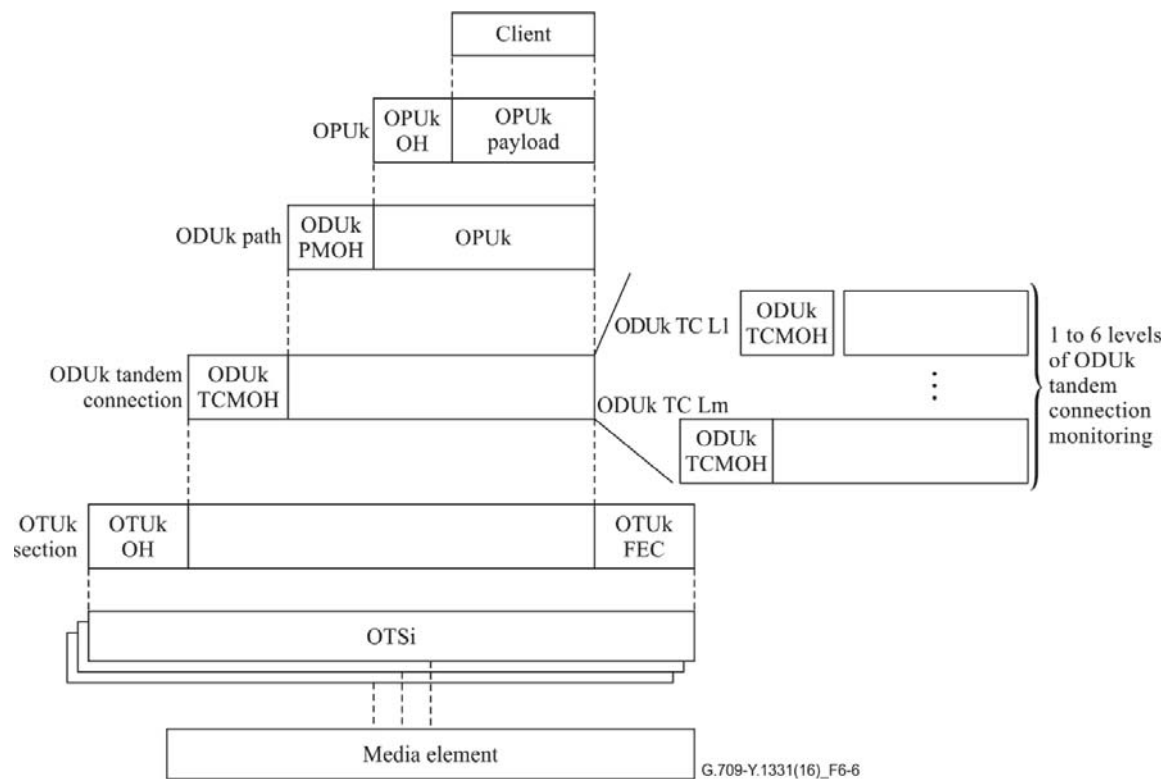


Figure 6-6 – MOTU interface principal information containment relationships

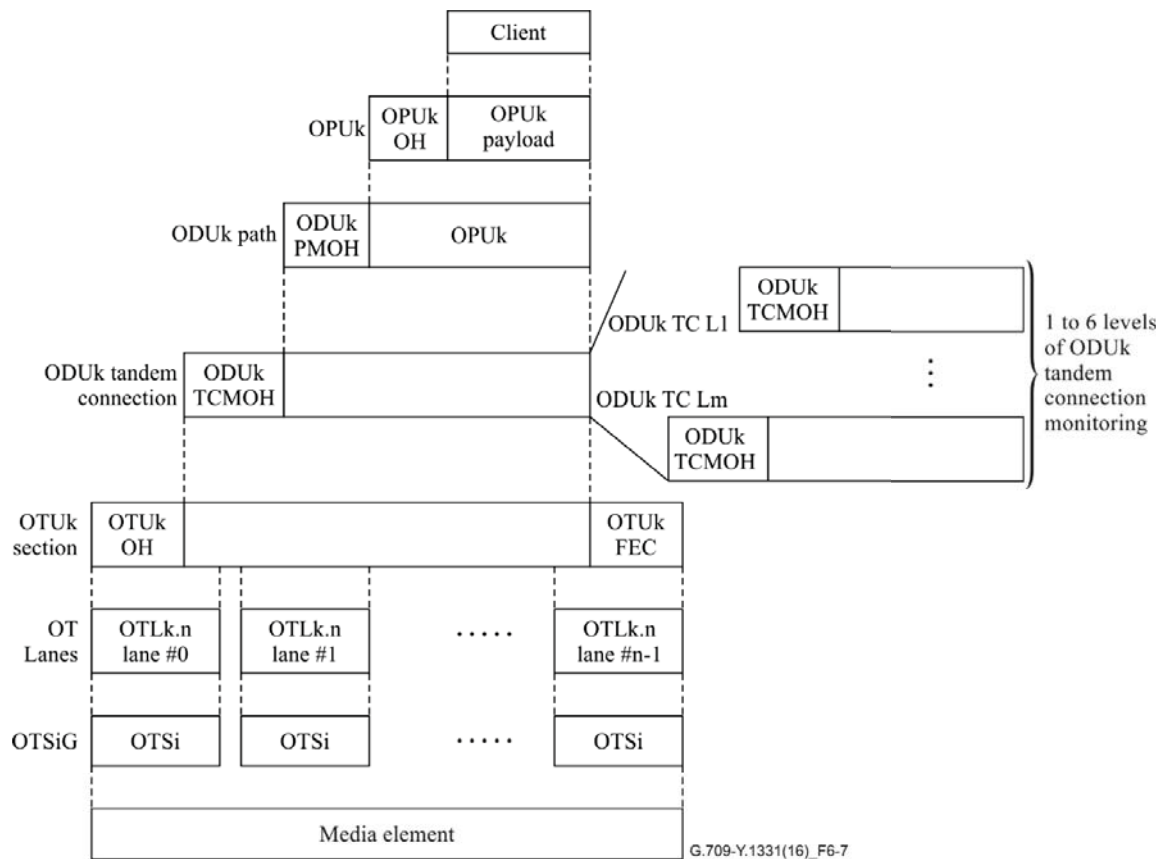


Figure 6-7 – Multi-lane SOTU interface principal information containment relationships

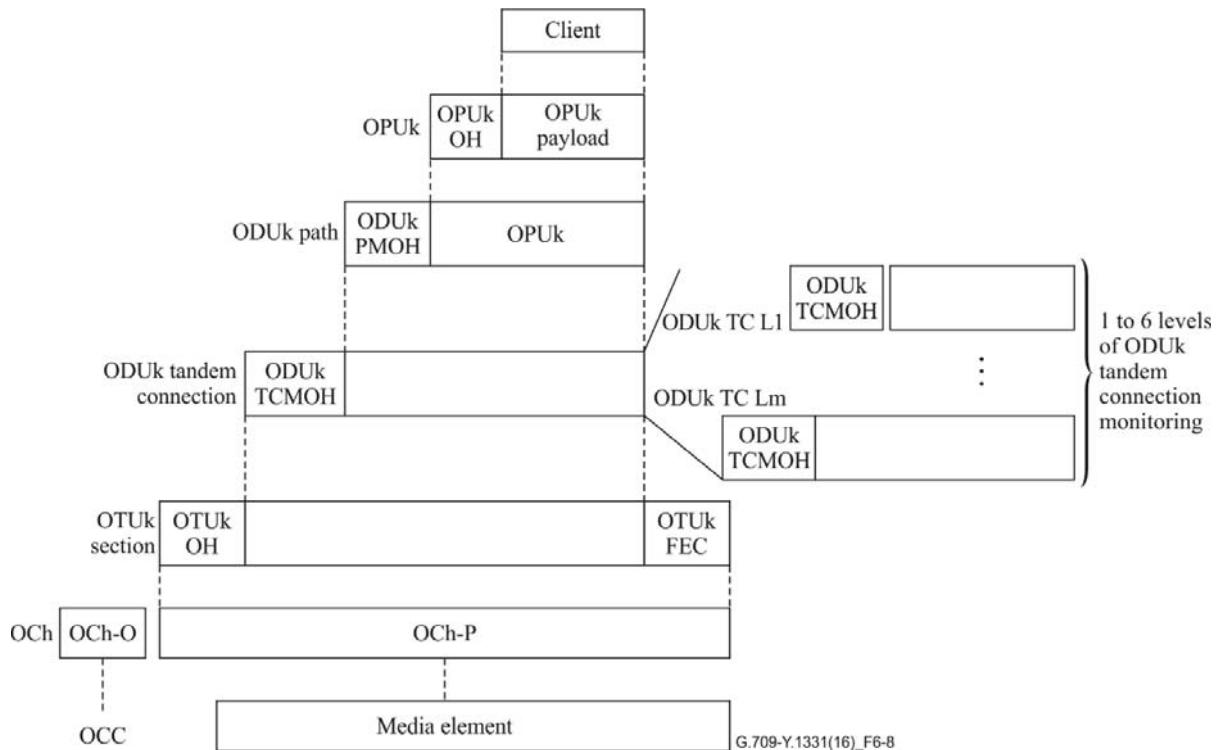


Figure 6-8 – SOTUm interface principal information containment relationships

7 Multiplexing/mapping principles and bit rates

Figure 7-1 shows the relationship between various information structure elements and illustrates the multiplexing structure and mappings for the OTU. In the multi-domain OTN any combination of the ODU multiplexing layers may be present at a given OTN NNI. The interconnection of and visibility of ODU multiplexing layers within an equipment or domain is outside the scope of this Recommendation. Figure 7-1 shows that a (non-OTN) client signal is mapped into an OPU. This OPU signal is mapped into the associated ODU. This ODU signal is either mapped into the associated OTU[V] signal, or into an ODTU. This ODTU signal is multiplexed into an ODTU Group (ODTUG). The ODTUG signal is mapped into an OPU. This OPU signal is mapped into the associated ODU, etc.

The OPU_k (k=0,1,2,3,4,flex) are the same information structures, but with different client signals. The OPU_{Cn} has a different information structure than the OPU_k; the OPU_{Cn} information structure consists of n times the information structure of the OPU while the OPU_k contain a single instance of the OPU information structure.

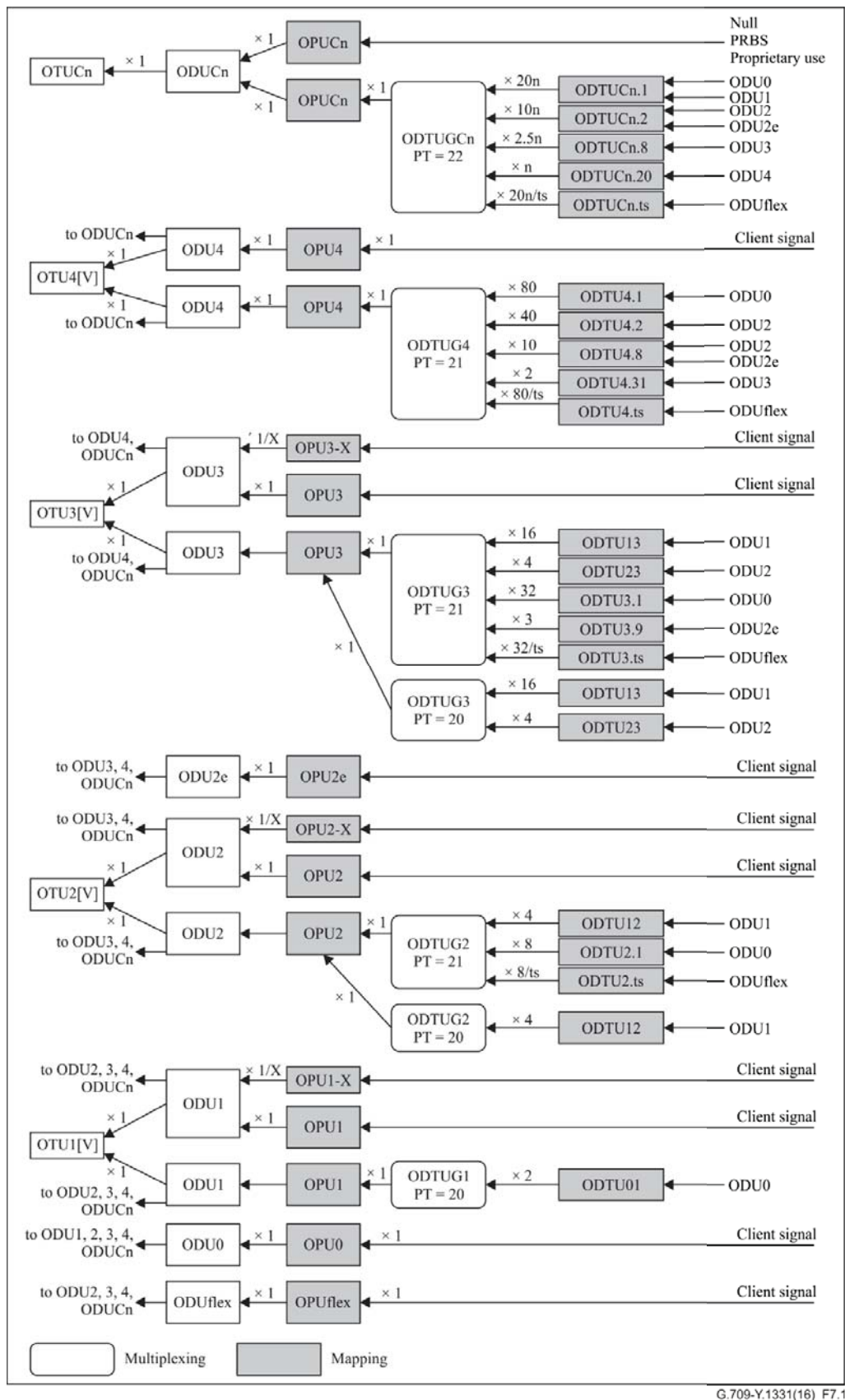


Figure 7-1 – OTN multiplexing and mapping structures

7.1 Mapping

The client signal or an optical data tributary unit group (ODTUG) is mapped into the OPU. The OPU is mapped into an ODU and the ODU is mapped into an OTU. The OTU is mapped into an OCh-P or OTSiG. The OTUk may also be mapped into an OTLk.n and an OTLk.n is then mapped into an OTSiG.

7.2 Wavelength division multiplex

Up to n ($n \geq 1$) OCh-P/OTSiG are multiplexed into an OMS-P or OPS using wavelength division multiplexing for transport over a MOTUm interface or MOTU interface.

For the case of the MOTUm interfaces the OSC is multiplexed into the MOTUm interface using wavelength division multiplexing.

n OTSi are aggregated into n frequency slots in an OPS-P using wavelength division multiplexing for transport over a multi-lane SOTU interface.

7.3 Bit rates and capacity

The bit rates and tolerance of the OTU signals are defined in Table 7-1.

The bit rates and tolerance of the ODU signals are defined in clause 12.2 and Table 7-2.

The bit rates and tolerance of the OPU payload are defined in Table 7-3.

The OTU, ODU and OPU frame periods are defined in Table 7-4.

The types and bit rates of the OTL signals are defined in Table 7-5.

The 2.5G and 1.25G tributary slot related OPUk multiframe periods and 5G tributary slot OPUCn multiframe periods are defined in Table 7-6.

The ODTU payload area bandwidths are defined in Table 7-7. The bandwidth depends on the OPU type and the mapping procedure (AMP or GMP). The AMP bandwidths include the bandwidth provided by the NJO overhead byte. GMP is defined without such NJO bytes.

The bit rates and tolerance of the ODUFlex(GFP) are defined in Table 7-8.

The number of OPU tributary slots required by a client ODU are summarized in Table 7-9 and specified in clauses 19.6 and 20.5.

Table 7-1 – OTU types and bit rates

OTU type	OTU nominal bit rate	OTU bit-rate tolerance
OTU1	$255/238 \times 2\,488\,320$ kbit/s	±20 ppm
OTU2	$255/237 \times 9\,953\,280$ kbit/s	
OTU3	$255/236 \times 39\,813\,120$ kbit/s	
OTU4	$255/227 \times 99\,532\,800$ kbit/s	
OTUCn	$n \times 239/226 \times 99\,532\,800$ kbit/s	
NOTE 1 – The nominal OTU rates are approximately: 2 666 057.143 kbit/s (OTU1), 10 709 225.316 kbit/s (OTU2), 43 018 413.559 kbit/s (OTU3), 111 809 973.568 kbit/s (OTU4) and $n \times 105\,258\,138.053$ kbit/s (OTUCn).		
NOTE 2 – OTU0, OTU2e and OTUflex are not specified in this Recommendation. ODU0 signals are to be transported over ODU1, ODU2, ODU3, ODU4 or ODUCn signals, ODU2e signals are to be transported over ODU3, ODU4 and ODUCn signals and OTUflex signals are transported over ODU2, ODU3, ODU4 and ODUCn signals.		
NOTE 3 – The OTUk (k=1,2,3,4) signal bit rates include the FEC overhead area. The OTUCn signal bit rates do not include a FEC overhead area.		

Table 7-2 – ODU types and bit rates

ODU type	ODU nominal bit rate	ODU bit-rate tolerance
ODU0	1 244 160 kbit/s	±20 ppm
ODU1	$239/238 \times 2\,488\,320$ kbit/s	
ODU2	$239/237 \times 9\,953\,280$ kbit/s	
ODU3	$239/236 \times 39\,813\,120$ kbit/s	
ODU4	$239/227 \times 99\,532\,800$ kbit/s	
ODUCn	$n \times 239/226 \times 99\,532\,800$ kbit/s	
ODU2e	$239/237 \times 10\,312\,500$ kbit/s	±100 ppm
ODUflex for CBR client signals	$239/238 \times$ client signal bit rate	±100 ppm (Notes 2, 3)
ODUflex for GFP-F mapped client signals	Configured bit rate (see Table 7-8)	±100 ppm
ODUflex for IMP mapped client signals	$s \times 239/238 \times 5\,156\,250$ kbit/s $s = 2, 8, n \times 5$ with $n \geq 1$ (Note 4)	±100 ppm
ODUflex for FlexE-aware client signals	$103\,125\,000 \times 240/238 \times n/20$ kbit/s ($n = n_1 + n_2 + \dots + n_p$)	± 100 ppm
<p>NOTE 1 – The nominal ODU rates are approximately: 2 498 775.126 kbit/s (ODU1), 10 037 273.924 kbit/s (ODU2), 40 319 218.983 kbit/s (ODU3), 104 794 445.815 kbit/s (ODU4), 10 399 525.316 kbit/s (ODU2e), $n \times 105\,258\,138.053$ kbit/s (ODUCn).</p> <p>NOTE 2 – The bit-rate tolerance for ODUflex(CBR) signals is specified as ±100 ppm. This value may be larger than the tolerance for the client signal itself (e.g., ±20 ppm). For such case, the tolerance is determined by the ODUflex(CBR) maintenance signals, which have a tolerance of ±100 ppm.</p> <p>NOTE 3 – For ODUflex(CBR) signals with nominal bit rates close to the maximum ODTUk.ts payload bit rate and client rate tolerances less than ±100 ppm (e.g., ±10 ppm), the ODUflex(CBR) maintenance signal bit rates may exceed the ODTUk.ts payload bit rate. For such cases either an additional tributary slot may be used (i.e., ODTUk.(ts+1)), or the nominal bit rate of the ODUflex(CBR) signal may be artificially reduced to a value of 100 ppm below the maximum ODUflex(CBR) signal bit rate.</p> <p>NOTE 4 – Refer to clause 12.2.6 for considerations on the values of "s".</p>		

Table 7-3 – OPU types and bit rates

OPU type	OPU payload nominal bit rate	OPU payload bit-rate tolerance
OPU0	$238/239 \times 1\,244\,160$ kbit/s	±20 ppm
OPU1	2 488 320 kbit/s	
OPU2	$238/237 \times 9\,953\,280$ kbit/s	
OPU3	$238/236 \times 39\,813\,120$ kbit/s	
OPU4	$238/227 \times 99\,532\,800$ kbit/s	
OPUCn	$n \times 238/226 \times 99\,532\,800$ kbit/s	
OPU2e	$238/237 \times 10\,312\,500$ kbit/s	±100 ppm
OPUflex for CBR client signals	client signal bit rate	client signal bit-rate tolerance, with a maximum of ±100 ppm

Table 7-3 – OPU types and bit rates

OPU type	OPU payload nominal bit rate	OPU payload bit-rate tolerance
OPUflex for GFP-F mapped client signals	$238/239 \times \text{ODUflex signal rate}$	± 100 ppm
OPUflex for IMP mapped client signals	$s \times 5 \ 156 \ 250$ kbit/s $s = 2, 8, n \times 5$ with $n \geq 1$ (Note 2)	± 100 ppm
OPUflex for FlexE-aware client signals	$103 \ 125 \ 000 \times 240/239 \times n/20$ kbit/s ($n = n_1 + n_2 + \dots + n_p$)	± 100 ppm
NOTE 1 – The nominal OPU payload rates are approximately: 1 238 954.310 kbit/s (OPU0 Payload), 2 488 320.000 kbit/s (OPU1 Payload), 9 995 276.962 kbit/s (OPU2 Payload), 40 150 519.322 kbit/s (OPU3 Payload), 104 355 975.330 (OPU4 Payload), 10 356 012.658 kbit/s (OPU2e Payload), $n \times 104 \ 817 \ 727.434$ kbit/s (OPUCn Payload). NOTE 2 – Refer to 12.2.6 for considerations on the values of "s".		

Table 7-4 – OTU/ODU/OPU frame periods

OTU/ODU/OPU type	Period (Note)
ODU0/OPU0	98.354 μ s
OTU1/ODU1/OPU1	48.971 μ s
OTU2/ODU2/OPU2	12.191 μ s
OTU3/ODU3/OPU3	3.035 μ s
OTU4/ODU4/OPU4	1.168 μ s
ODU2e/OPU2e	11.767 μ s
OTUCn/ODUCn/OPUCn	1.163 μ s
ODUflex/OPUflex	CBR client signals: 121856/client_signal_bit_rate
	GFP-F mapped client signals: 122368/ODUflex_bit_rate
	IMP mapped client signals: 122368/ODUflex_bit_rate
	FlexE-aware client signals: 122368/ODUflex_bit_rate
NOTE – The period is an approximated value, rounded to 3 decimal places.	

Table 7-5 – OTL types and bit rates

OTL type	OTL nominal bit rate	OTL bit-rate tolerance
OTL3.4	4 lanes of $255/236 \times 9\,953\,280$ kbit/s	±20 ppm
OTL4.4	4 lanes of $255/227 \times 24\,883\,200$ kbit/s	
NOTE – The nominal OTL rates are approximately: 10 754 603.390 kbit/s (OTL3.4) and 27 952 493.392 kbit/s (OTL4.4).		

Table 7-6 – OPUk multiframe periods for 2.5G and 1.25G tributary slots and ODU_{Cn} multiframe periods for 5G tributary slots

OPU type	1.25G tributary slot multiframe period (Note)	2.5G tributary slot multiframe period (Note)	5G tributary slot multiframe period (Note)
OPU1	97.942 μ s	–	–
OPU2	97.531 μ s	48.765 μ s	–
OPU3	97.119 μ s	48.560 μ s	–
OPU4	93.416 μ s	–	–
OPUC _n	–	–	23.251 μ s
NOTE – The period is an approximated value, rounded to 3 decimal places.			

Table 7-7 – ODTU payload bandwidth (kbit/s)

ODTU type	ODTU payload nominal bandwidth		ODTU payload bit-rate tolerance
ODTU01	(1904 + 1/8)/3824 × ODU1 bit rate		±20 ppm
ODTU12	(952 + 1/16)/3824 × ODU2 bit rate		
ODTU13	(238 + 1/64)/3824 × ODU3 bit rate		
ODTU23	(952 + 4/64)/3824 × ODU3 bit rate		
ODTU2.ts	ts × 476/3824 × ODU2 bit rate		
ODTU3.ts	ts × 119/3824 × ODU3 bit rate		
ODTU4.ts	ts × 47.5/3824 × ODU4 bit rate		
ODTUCn.ts	ts × 190.4/3824 × ODUCn bit rate/n		
	Minimum	Nominal	Maximum
ODTU01	1 244 216.796	1 244 241.681	1 244 266.566
ODTU12	2 498 933.311	2 498 983.291	2 499 033.271
ODTU13	2 509 522.012	2 509 572.203	2 509 622.395
ODTU23	10 038 088.048	10 038 288.814	10 038 489.579
ODTU2.ts	ts × 1 249 384.632	ts × 1 249 409.620	ts × 1 249 434.608
ODTU3.ts	ts × 1 254 678.635	ts × 1 254 703.729	ts × 1 254 728.823
ODTU4.ts	ts × 1 301 683.217	ts × 1 301 709.251	ts × 1 301 735.285
ODTUCn.ts	ts × 5 240 781.554	ts × 5 240 886.372	ts × 5 240 991.189
NOTE – The bandwidth is an approximated value, rounded to 3 decimal places.			

Table 7-8 – Recommended ODUflex (GFP) bit rates and tolerance

ODU type	Nominal bit-rate	Tolerance
ODU2.ts (Note 1)	1'249'177.230 kbit/s	
ODU3.ts (Note 1)	1'254'470.354 kbit/s	
ODU4.ts (Note 1)	1'301'467.133 kbit/s	
ODUflex(GFP) of n 1.25G tributary slots, $1 \leq n \leq 8$	$n \times \text{ODU2.ts}$	± 100 ppm
ODUflex(GFP) of n 1.25G tributary slots, $9 \leq n \leq 32$	$n \times \text{ODU3.ts}$	± 100 ppm
ODUflex(GFP) of n 1.25G tributary slots, $33 \leq n \leq 80$ (Note 2)	$n \times \text{ODU4.ts}$	± 100 ppm
NOTE 1 – The values of ODUk.ts are chosen to permit a variety of methods to be used to generate an ODUflex(GFP) clock. See Appendix XI for the derivation of these values and example ODUflex(GFP) clock generation methods.		
NOTE 2 – Transport of packet clients via an ODUflex(GFP) is specified up to 100 Gbit/s.		

Table 7-9 – Number of tributary slots required for ODUj into OPUk and for ODUk into OPUCn

	# 5G TS	# 2.5G tributary slots		# 1.25G tributary slots			
	OPUCn	OPU2	OPU3	OPU1	OPU2	OPU3	OPU4
ODU0	1	–	–	1	1	1	1
ODU1	1	1	1	–	2	2	2
ODU2	2	–	4	–	–	8	8
ODU2e	2	–	–	–	–	9	8
ODU3	8	–	–	–	–	–	31
ODUflex(CBR)							
– ODUflex(IB SDR)	1	–	–	–	3	3	2
– ODUflex(IB DDR)	1	–	–	–	5	4	4
– ODUflex(IB QDR)	2	–	–	–	–	9	8
– ODUflex(FC-400)	1	–	–	–	4	4	4
– ODUflex(FC-800)	2	–	–	–	7	7	7
– ODUflex(FC-1600)	3	–	–	–	–	12	11
– ODUflex(FC-3200)	6	–	–	–	–	23	22
– ODUflex(3G SDI) (2 970 000)	1	–	–	–	3	3	3
– ODUflex(3G SDI) (2 970 000/1.001)	1	–	–	–	3	3	3
– ODUflex(GFP)	Note	–	–	–	n	n	n
– ODUflex(IMP)	Note	–	–	–	–	Note	Note
– ODUflex(FlexE)	Note	–	–	–	–	Note	Note
NOTE – Refer to equations 19-1a, 19-1b, 20-1a and 20-1b in clauses 19.6 and 20.5.							

7.4 ODUk time-division multiplex

Figure 7-1 shows the relationship between various time-division multiplexing elements that are defined below and illustrates possible multiplexing structures. Table 7-10 provides an overview of valid tributary slot types and mapping procedure configuration options.

Up to 2 ODU0 signals are multiplexed into an ODTUG1 (PT=20) using time-division multiplexing. The ODTUG1 (PT=20) is mapped into the OPU1.

Up to 4 ODU1 signals are multiplexed into an ODTUG2 (PT=20) using time-division multiplexing. The ODTUG2 (PT=20) is mapped into the OPU2.

A mixture of p ($p \leq 4$) ODU2 and q ($q \leq 16$) ODU1 signals can be multiplexed into an ODTUG3 (PT=20) using time-division multiplexing. The ODTUG3 (PT=20) is mapped into the OPU3.

A mixture of p ($p \leq 8$) ODU0, q ($q \leq 4$) ODU1, r ($r \leq 8$) ODUflex signals can be multiplexed into an ODTUG2 (PT=21) using time-division multiplexing. The ODTUG2 (PT=21) is mapped into the OPU2.

A mixture of p ($p \leq 32$) ODU0, q ($q \leq 16$) ODU1, r ($r \leq 4$) ODU2, s ($s \leq 3$) ODU2e and t ($t \leq 32$) ODUflex signals can be multiplexed into an ODTUG3 (PT=21) using time-division multiplexing. The ODTUG3 (PT=21) is mapped into the OPU3.

A mixture of p ($p \leq 80$) ODU0, q ($q \leq 40$) ODU1, r ($r \leq 10$) ODU2, s ($s \leq 10$) ODU2e, t ($t \leq 2$) ODU3 and u ($u \leq 80$) ODUflex signals can be multiplexed into an ODTUG4 (PT=21) using time-division multiplexing. The ODTUG4 (PT=21) is mapped into the OPU4.

NOTE 1 – The ODTUG k is a logical construct and is not defined further. ODTU jk and ODTU k .ts signals are directly time-division multiplexed into the tributary slots of an OPU k .

A mixture of p ($p \leq 10n$) ODU0, q ($q \leq 10n$) ODU1, r ($r \leq 10n$) ODU2, s ($s \leq 10n$) ODU2e, t ($t \leq \text{int}(10n/4)$) ODU3, u ($u \leq n$) ODU4 and v ($v \leq 10n$) ODUflex signals can be multiplexed into an ODTUGC n (PT=22) using time-division multiplexing. The ODTUGC n (PT=22) is mapped into the OPUC n .

NOTE 2 – The ODTUG k and ODTUGC n are logical constructs and are not defined further. ODTU jk and ODTU k .ts signals are directly time-division multiplexed into the tributary slots of an OPU k . ODTUC n .ts signals are directly time-division multiplexed into the tributary slots of an OPUC n .

NOTE 3 – Implementations should support the multiplexing of up to $10n$ ODU k ($k=0,1,2,2e,3,4,\text{flex}$) signals into an OPUC n . Support for the multiplexing of up to $20n$ ODU k ($k=0,1,\text{flex}$) signals into the OPUC n is not required.

Table 7-10 – Overview of ODU j into OPU k and ODU k into OPUC n mapping types

	5G tributary slots	2.5G tributary slots		1.25G tributary slots			
	OPUC n (PT=22)	OPU2 (PT=20)	OPU3 (PT=20)	OPU1 (PT=20)	OPU2 (PT=21)	OPU3 (PT=21)	OPU4 (PT=21)
ODU0	GMP - Note	–	–	AMP	GMP	GMP	GMP
ODU1	GMP - Note	AMP	AMP	–	AMP	AMP	GMP
ODU2	GMP	–	AMP	–	–	AMP	GMP
ODU2e	GMP	–	–	–	–	GMP	GMP
ODU3	GMP	–	–	–	–	–	GMP
ODU4	GMP	–	–	–	–	–	–
ODUflex	GMP	–	–	–	GMP	GMP	GMP
NOTE – Mapping ODU0 and ODU1 into a 5G tributary slot of the OPUC n does not fully occupy the tributary slot's bandwidth.							

Figures 7-2, 7-3 and 7-4 show how various signals are multiplexed using the ODTUG1/2/3 (PT=20) multiplexing elements. Figure 7-2 presents the multiplexing of four ODU1 signals into the OPU2 signal via the ODTUG2 (PT=20). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 1 into 2 (ODTU12) using the AMP justification overhead (JOH). The four ODTU12 signals are time-division multiplexed into the optical data tributary unit group 2 (ODTUG2) with payload type 20, after which this signal is mapped into the OPU2.

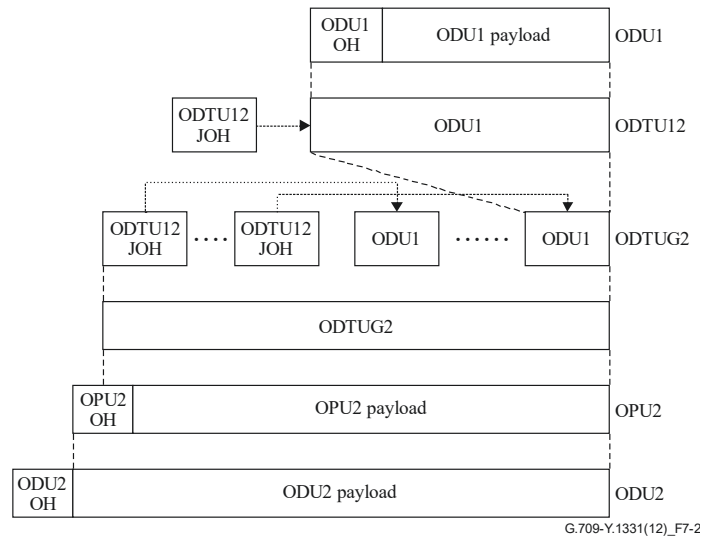


Figure 7-2 – ODU1 into ODU2 multiplexing method via ODTUG2 (PT=20)

Figure 7-3 presents the multiplexing of up to 16 ODU1 signals and/or up to 4 ODU2 signals into the OPU3 signal via the ODTUG3 (PT=20). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 1 into 3 (ODTU13) using the AMP justification overhead (JOH). An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 2 into 3 (ODTU23) using the AMP justification overhead (JOH). "x" ODTU23 ($0 \leq x \leq 4$) signals and "16-4x" ODTU13 signals are time-division multiplexed into the optical data tributary unit group 3 (ODTUG3) with payload type 20, after which this signal is mapped into the OPU3.

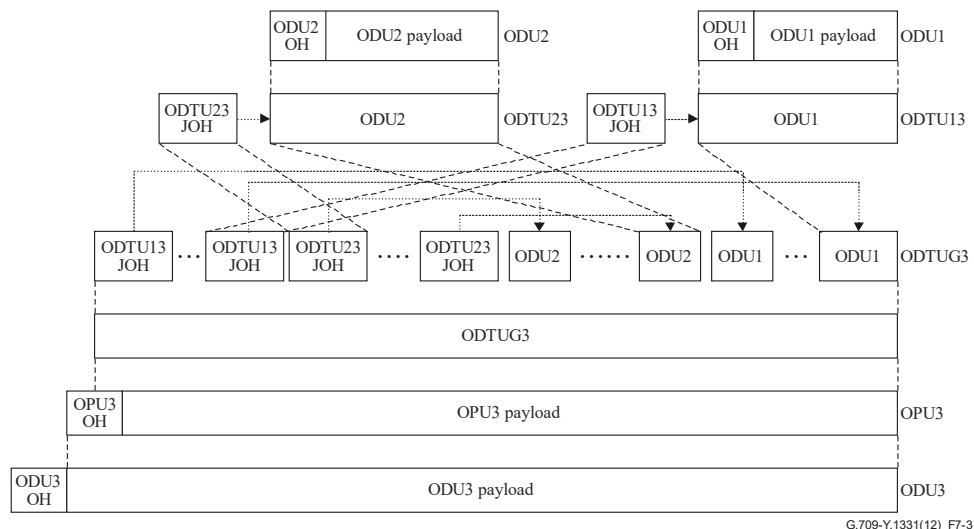


Figure 7-3 – ODU1 and ODU2 into ODU3 multiplexing method via ODTUG3 (PT=20)

Figure 7-4 presents the multiplexing of two ODU0 signals into the OPU1 signal via the ODTUG1 (PT=20). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 0 into 1 (ODTU01) using the AMP justification overhead (JOH). The two ODTU01 signals are time-division multiplexed into the optical data tributary unit group 1 (ODTUG1) with payload type 20, after which this signal is mapped into the OPU1.

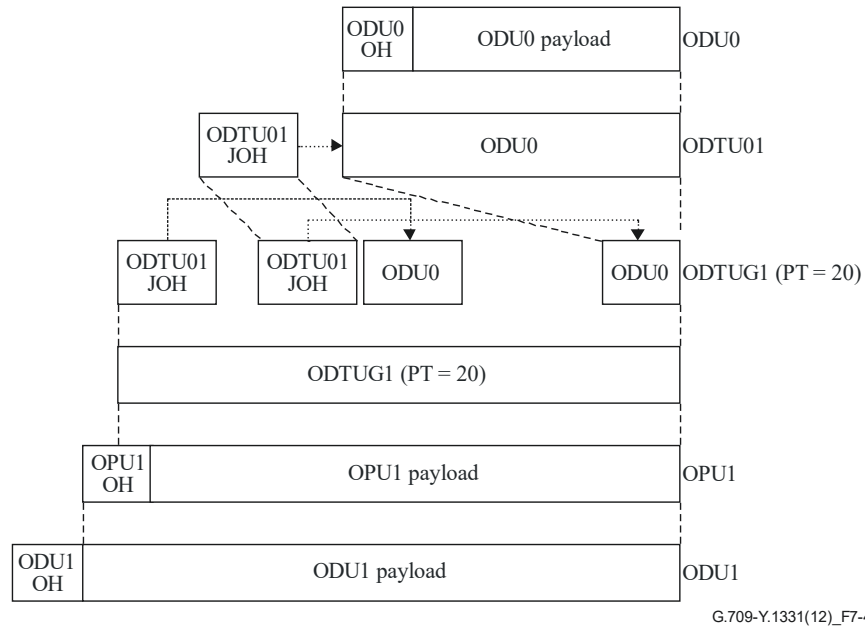


Figure 7-4 – ODU0 into ODU1 multiplexing method via ODTUG1 (PT=20)

Figures 7-5, 7-6 and 7-7 show how various signals are multiplexed using the ODTUG2/3/4 (PT=21) multiplexing elements.

Figure 7-5 presents the multiplexing of up to eight ODU0 signals, and/or up to four ODU1 signals and/or up to eight ODUFlex signals into the OPU2 signal via the ODTUG2 (PT=21). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 1 into 2 (ODTU12) using the AMP justification overhead (JOH). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 2.1 (ODTU2.1) using the GMP justification overhead. An ODUFlex signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 2.ts (ODTU2.ts) using the GMP justification overhead. Up to eight ODTU2.1 signals, up to four ODTU12 signals and up to eight ODTU2.ts signals are time-division multiplexed into the optical data tributary unit group 2 (ODTUG2) with payload type 21, after which this signal is mapped into the OPU2.

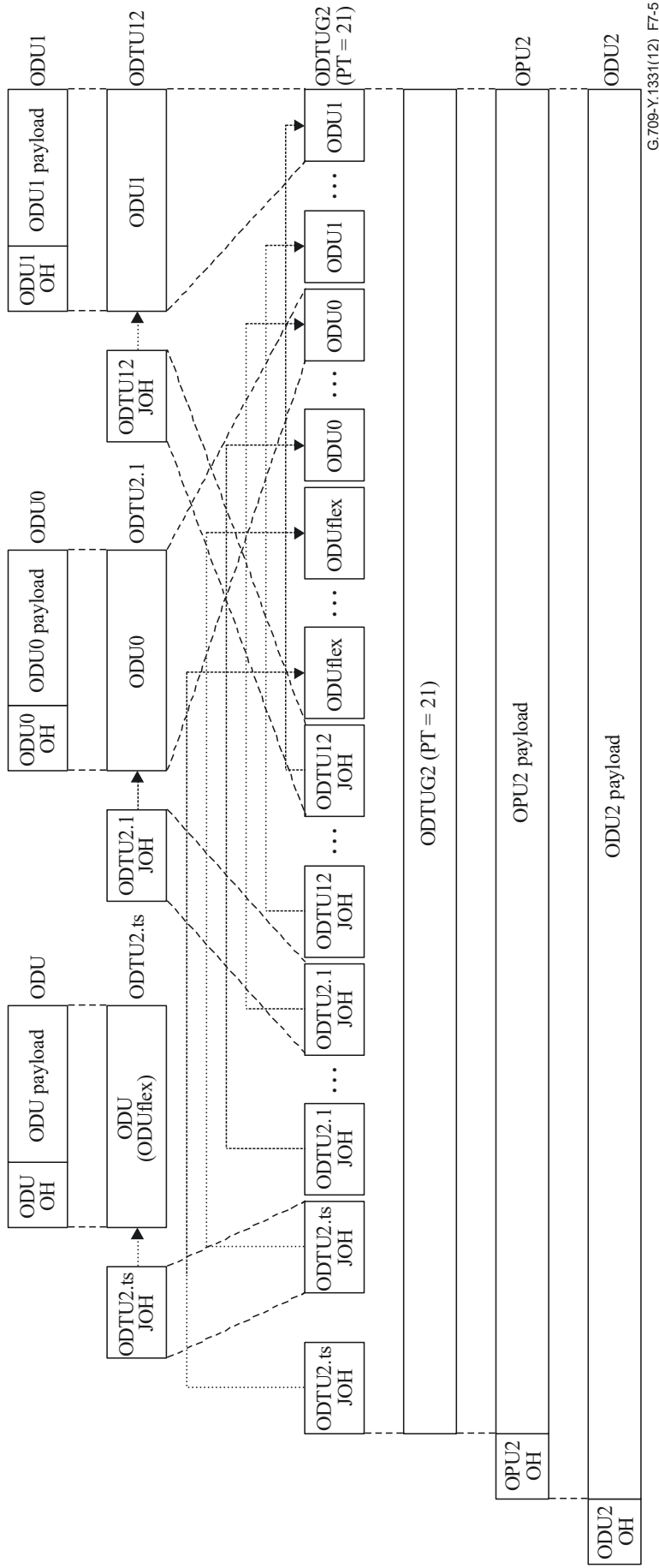


Figure 7-5 – ODU0, ODU1 and ODUflex into ODU2 multiplexing method via ODTUG2 (PT=21)

Figure 7-6 presents the multiplexing of up to thirty-two ODU0 signals and/or up to sixteen ODU1 signals and/or up to four ODU2 signals and/or up to three ODU2e signals and/or up to thirty-two ODUflex signals into the OPU3 signal via the ODTUG3 (PT=21). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 1 into 3 (ODTU13) using the AMP justification overhead (JOH). An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 2 into 3 (ODTU23) using the AMP justification overhead. An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 3.1 (ODTU3.1) using the GMP justification overhead. An ODU2e signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 3.9 (ODTU3.9) using the GMP justification overhead. An ODUflex signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 3.ts (ODTU3.ts) using the GMP justification overhead. Up to thirty-two ODTU3.1 signals, up to sixteen ODTU13 signals, up to four ODTU23 signals, up to three ODTU3.9 and up to thirty-two ODTU3.ts signals are time-division multiplexed into the optical data tributary unit group 3 (ODTUG3) with payload type 21, after which this signal is mapped into the OPU3.

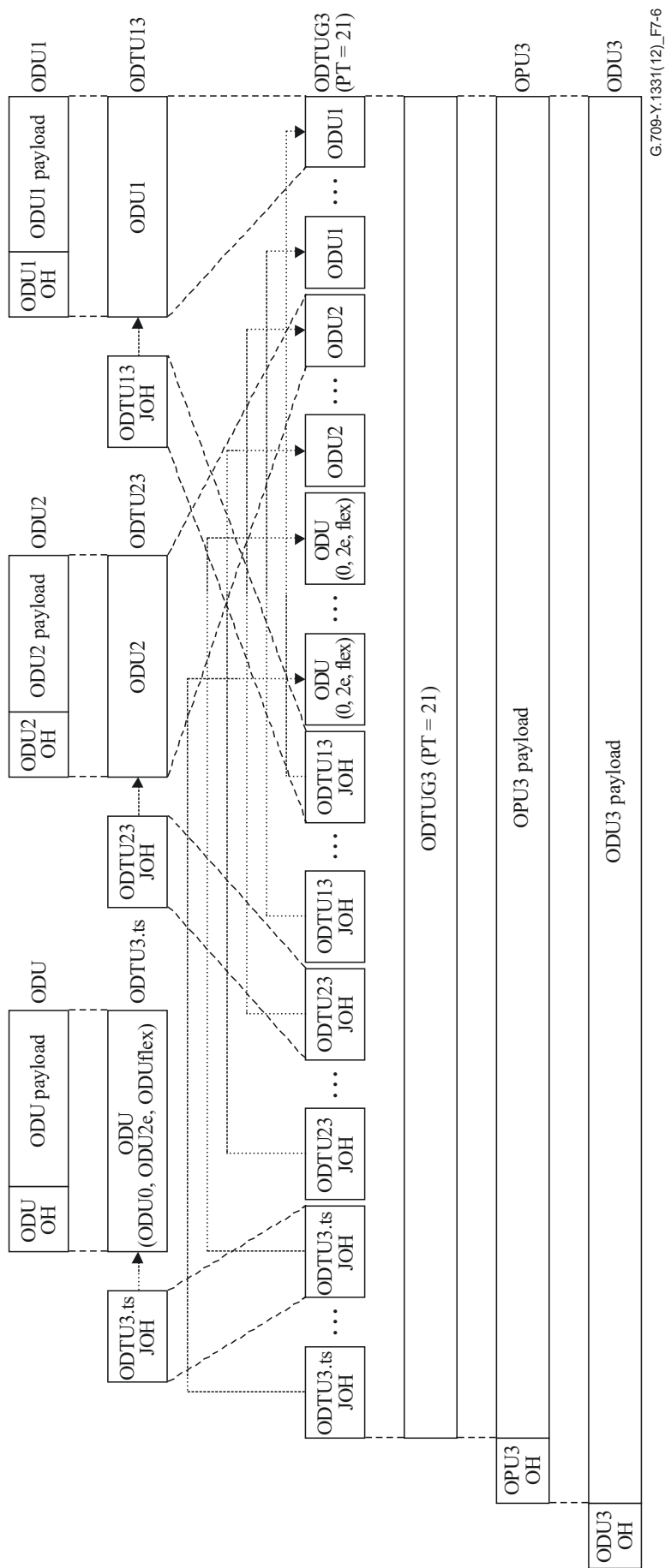
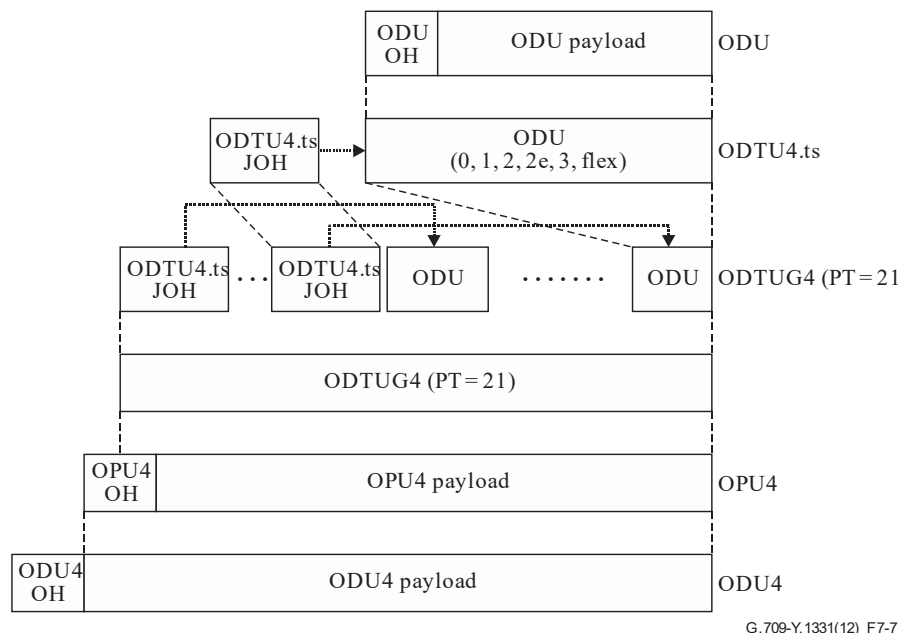


Figure 7-6 – ODU0, ODU1, ODU2, ODU2e and ODUflex into ODU3 multiplexing method via ODTUG3 (PT=21)

Figure 7-7 presents the multiplexing of up to eighty ODU0 signals and/or up to forty ODU1 signals and/or up to ten ODU2 signals and/or up to ten ODU2e signals and/or up to two ODU3 signals and/or up to eighty ODUFlex signals into the OPU4 signal via the ODTUG4 (PT=21). An ODU0 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.1 (ODTU4.1) using the GMP justification overhead (JOH). An ODU1 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.2 (ODTU4.2) using the GMP justification overhead. An ODU2 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.8 (ODTU4.8) using the GMP justification overhead (JOH). An ODU2e signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.8 (ODTU4.8) using the GMP justification overhead. An ODU3 signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.31 (ODTU4.31) using the GMP justification overhead. An ODUFlex signal is extended with a frame alignment overhead and asynchronously mapped into the optical data tributary unit 4.ts (ODTU4.ts) using the GMP justification overhead (JOH). Up to eighty ODTU4.1 signals, up to forty ODTU4.2 signals, up to ten ODTU4.8 signals, up to two ODTU4.31 and up to eighty ODTU4.ts signals are time-division multiplexed into the optical data tributary unit group 4 (ODTUG4) with payload type 21, after which this signal is mapped into the OPU4.



G.709-Y.1331(12) F7-7

Figure 7-7 – ODU0, ODU1, ODU2, ODU2e, ODU3 and ODUFlex into ODU4 multiplexing method via ODTUG4 (PT=21)

Figure 7-8 presents the multiplexing of up to $10n$ ODU0 signals and/or up to $10n$ ODU1 signals and/or up to $10n$ ODU2 signals and/or up to $10n$ ODU2e signals and/or up to $\text{int}(10n/4)$ ODU3 signals and/or up to n ODU4 signals and/or up to $10n$ ODUFlex signals into the OPUCn signal via the ODTUGCn (PT=22). An ODU k signal is extended with frame alignment overhead and asynchronously mapped into the Optical Data Tributary Unit Cn.ts (ODTUCn.ts) ($\langle k, ts \rangle = \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 2e, 2 \rangle, \langle 3, 8 \rangle, \langle 4, 20 \rangle, \langle \text{flex}, ts \rangle$) using the GMP justification overhead (JOH). Up to $10n$ ODTUCn.1 signals, up to $\text{int}(10n/4)$ ODTUCn.4 signals, up to n ODTUCn.10 signals and up to $10n$ ODTUCn.ts signals are time-division multiplexed into the Optical Data Tributary Unit Group Cn (ODTUGCn) with Payload Type 22, after which this signal is mapped into the OPUCn.

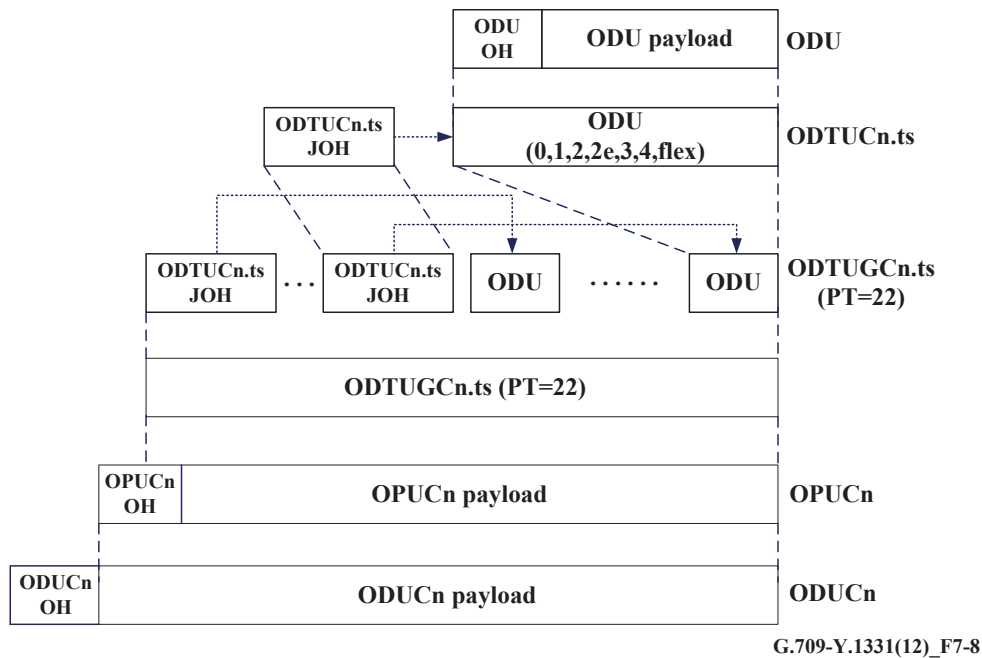


Figure 7-8 – ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4 and ODUflex into ODUcN multiplexing method via ODTUGcN (PT=22)

Details of the multiplexing method and mappings are given in clause 19 for OPUCn and clause 20 for ODUcN.

Some examples illustrating the multiplexing of 2 ODU0 signals into an ODU1 and of 4 ODU1 signals into an ODU2 are presented in Appendix III.

8 OTN Interfaces

8.1 Single-OTU (SOTU) interface

The SOTU interface supports one OTU on one OTSiG on a single optical span with 3R regeneration at each end. The OTSiG is composed of one or more OTSi.

Application codes of the optical tributary signal carrying the OTUk are contained in [ITU-T G.959.1] and [ITU-T G.693].

Application codes of the optical tributary signal carrying an OTLk.n lane are contained in [ITU-T G.959.1] and [ITU-T G.695].

Vendor specific application identifiers of the optical tributary signals or optical tributary signal groups carrying these OTUs are outside the scope of this Recommendation.

8.2 Multi-OTU (MOTU) interface

The MOTU interface supports n OTUs on n OTSiG on a single optical span with 3R regeneration at each end. Each OTSiG is composed of one or more OTSi.

At least one of the OTU signals is present during normal operation.

There is no predefined order in which the OTU signals are taken into service.

NOTE – MOTU interface overhead is not defined. The interface will use the OTU SMOH in this multi-channel interface for supervision and management. MOTU interface connectivity (TIM) failure reports will be computed from the individual OTU reports by means of failure correlation in fault management. Refer to the equipment Recommendations for further details.

Application codes of the optical tributary signal carrying an OTUk are contained in [ITU-T G.959.1] and [ITU-T G.695].

Vendor specific application identifiers of the optical tributary signals or optical tributary signal groups carrying these OTUs are outside the scope of this Recommendation.

8.3 Single-OTU with management (SOTUm) interface

The SOTUm interface supports one OTUk and non-associated overhead carried by an overhead communication channel (OCC). The OTUk is carried over an OTSi.

NOTE – The OTSi is referred to in other clauses as an OCh-P that carries an OTUk.

Application codes of the optical tributary signal carrying the OTUk are contained in [ITU-T G.698.1] and [ITU-T G.698.2].

Specifications of the OCC carrying the OCh-O are contained in [ITU-T G.7712].

8.4 Multi-OTU with management (MOTUm) interface

The MOTUm interface supports n ($n \geq 1$) OTUs on n OTSiG and non-associated overhead carried by an OSC or other means. 3R regeneration is not required at the interface.

NOTE – The OTSiG carrying an OTUk is referred to in other clauses as an OCh-P.

Vendor specific application identifiers of the OTSi carrying these OTUs are outside the scope of this Recommendation.

9 Media Element

A description of the media element will be provided in [ITU-T G.872].

10 OCh and OTSiA

The OCh and OTSiA transport a digital client signal between 3R regeneration points. The OCh and OTSiA client signals defined in this Recommendation are the OTUk, OTUk-v, OTUkV and OTUCn signals.

10.1 OCh

The OCh structure is conceptually shown in Figure 10-1. The OCh contains two parts: an overhead part (OCh-O) and a payload part (OCh-P).



Figure 10-1 – OCh information structure

10.2 Optical tributary signal assembly (OTSiA)

The OTSiA structure is conceptually shown in Figure 10-2. The OTSiA contains two parts: a payload part (OTSiG) and an overhead part (OTSiG-O).



Figure 10-2 – OTSiA information structure

11 Optical transport unit (OTU)

The OTUk[V] conditions the ODUk for transport over an OCh network connection. The OTUk frame structure, including the OTUk FEC is completely standardized. The OTUkV is a frame structure, including the OTUkV FEC that is only functionally standardized (i.e., only the required functionality is specified); refer to Appendix II. Besides these two, there is an OTUkV in which the completely standardized OTUk frame structure is combined with a functionally standardized OTUkV FEC; refer to appendix II. This combination is identified as OTUk-v.

The OTUCn frame structure is defined without an OTUCn FEC area. The FEC associated with an OTUCn is interface dependent, and specified as an element of each interface.

11.1 OTUk frame structure

The OTUk ($k = 1, 2, 3, 4$) frame structure is based on the ODUk frame structure and extends it with a forward error correction (FEC) as shown in Figure 11-1. 256 columns are added to the ODUk frame for the FEC and the reserved overhead bytes in row 1, columns 8 to 14 of the ODUk overhead are used for an OTUk specific overhead, resulting in an octet-based block frame structure with four rows and 4080 columns. The MSB in each octet is bit 1, the LSB is bit 8.

NOTE – This Recommendation does not specify an OTUk frame structure for $k=0$, $k=2e$ or $k=flex$. See Annex G for the specification of OTU0LL.

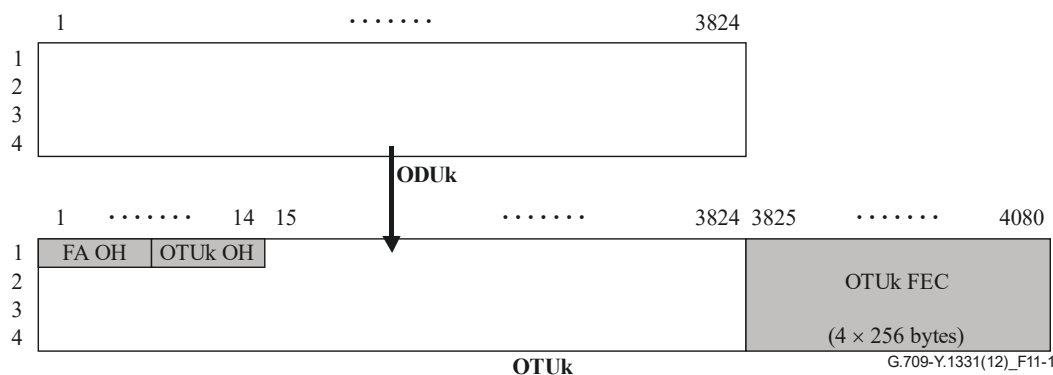


Figure 11-1 – OTUk frame structure

The bit rates of the OTUk signals are defined in Table 7-1.

The OTUk ($k=1, 2, 3, 4$) forward error correction (FEC) contains the Reed-Solomon RS(255,239) FEC codes. Transmission of the OTUk FEC is mandatory for $k=4$ and optional for $k=1, 2, 3$. If no FEC is transmitted, fixed stuff bytes (all-0s pattern) are to be used.

The RS(255,239) FEC code shall be computed as specified in Annex A.

For interworking of equipment supporting FEC, with equipment not supporting FEC (inserting fixed stuff all-0s pattern in the OTUk ($k=1, 2, 3$) FEC area), the FEC supporting equipment shall support the capability to disable the FEC decoding process (ignore the content of the OTUk ($k=1, 2, 3$) FEC).

The transmission order of the bits in the OTUk frame is left to right, top to bottom, and MSB to LSB (see Figure 11-2).

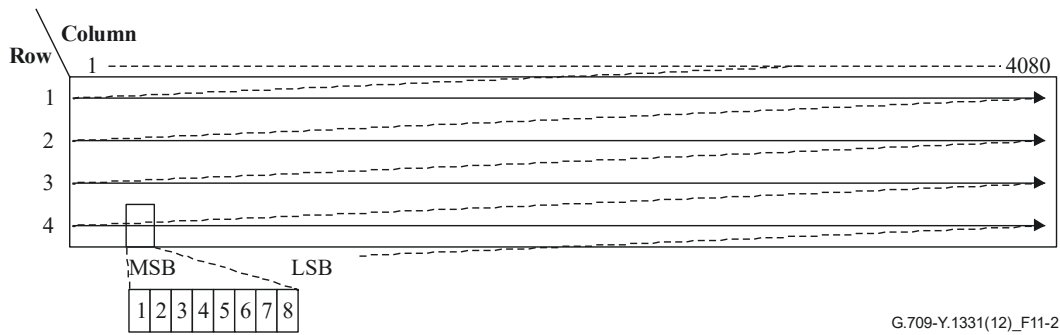


Figure 11-2 – Transmission order of the OTUk frame bits

11.2 Scrambling

The OTUk signal must have sufficient bit timing content at the ONNI. A suitable bit pattern, which prevents a long sequence of "1"s or "0"s, is provided by using a scrambler.

The operation of the scrambler shall be functionally identical to that of a frame synchronous scrambler of sequence length 65535 operating at the OTUk rate.

The generating polynomial shall be $1 + x + x^3 + x^{12} + x^{16}$. Figure 11-3 shows a functional diagram of the frame synchronous scrambler.

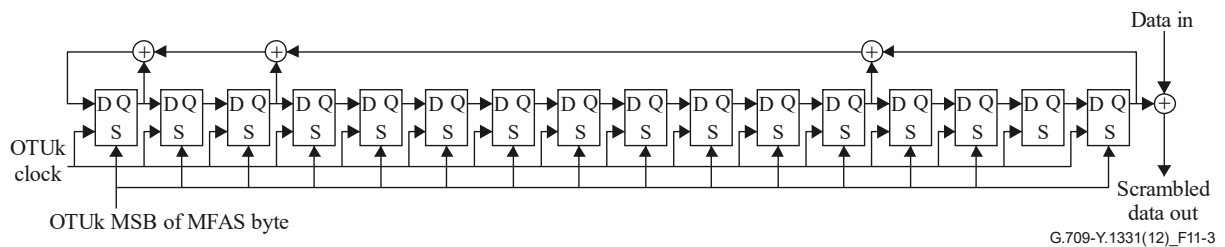


Figure 11-3 – Frame synchronous scrambler

The scrambler shall be reset to "FFFF" (HEX) on the most significant bit of the byte following the last framing byte in the OTUk frame, i.e., the MSB of the MFAS byte. This bit, and all subsequent bits to be scrambled shall be added modulo 2 to the output from the x^{16} position of the scrambler. The scrambler shall run continuously throughout the complete OTUk frame. The framing bytes (FAS) of the OTUk overhead shall not be scrambled.

Scrambling is performed after FEC computation and insertion into the OTUk signal.

11.3 OTUCn frame structure

The OTUCn frame structure (Figure 11-4) is based on the ODUCn frame structure and deploys the reserved overhead bytes in row 1, columns 8 to 14 of each ODU frame structure in the ODUCn overhead for an OTUCn specific overhead, resulting in an octet-based block frame structure with $n \times \text{four}$ rows and 3824 columns. The MSB in each octet is bit 1, the LSB is bit 8.

Interleaving of the n frame and multi-frame synchronous OTU frame structure instances within the OTUCn, forward error correction, encoding (e.g., scrambling), deskewing and transmission order of the OTUCn are interface specific and specified for inter-domain OTN interfaces with application codes in the interface specific Recommendations (ITU-T G.709.x series). For OTN interfaces with vendor specific application identifiers these specifications are vendor specific and out of scope of this Recommendation.

The bit rates of the OTUCn signals are defined in Table 7-1.

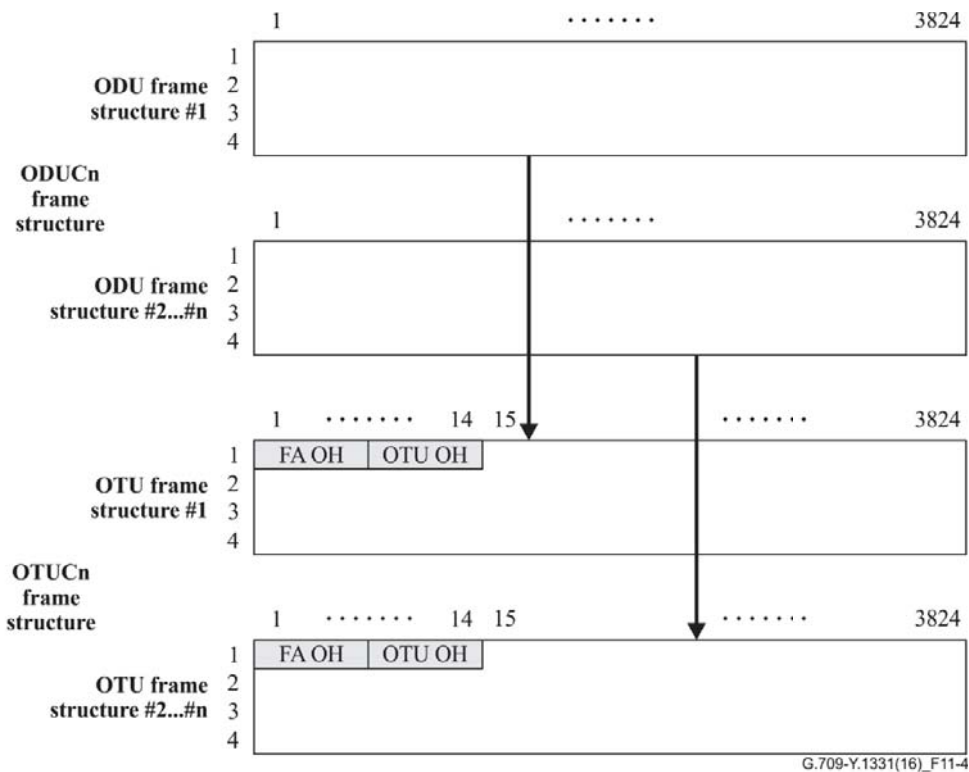


Figure 11-4– OTUCn frame structure

12 Optical data unit (ODU)

12.1 ODU frame structure

The ODU frame structure is shown in Figure 12-1. It is organized in an octet-based block frame structure with four rows and 3824 columns.

The ODU_k (k=0,1,2,2e,3,4,flex) frame structure contains one instance of the ODU frame structure. The ODU_{Cn} frame structure contains n frame and multi-frame synchronous instances of the ODU frame structures, numbered 1 to n (ODU #1 to ODU #n).

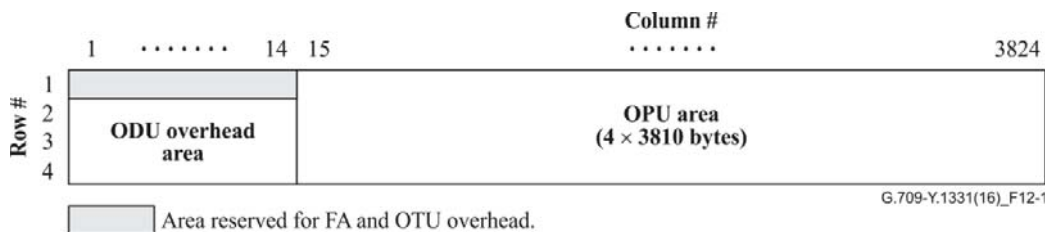


Figure 12-1 – ODU frame structure

The two main areas of the ODU frame are:

- ODU overhead area
- OPU area.

Columns 1 to 14 of the ODU are dedicated to ODU overhead area.

NOTE – Columns 1 to 14 of row 1 are reserved for a frame alignment and OTU specific overhead.

Columns 15 to 3824 of the ODU are dedicated to OPU area.

12.2 ODU bit rates and bit-rate tolerances

ODU_k signals may be generated using either a local clock, or the recovered clock of the client signal. In the latter case the ODU_k frequency and frequency tolerance are locked to the client signal's frequency and frequency tolerance. In the former case the ODU_k frequency and frequency tolerance are locked to the local clock's frequency and frequency tolerance. The local clock frequency tolerance for the OTN is specified to be ± 20 ppm.

ODUC_n signals are generated using a local clock. The ODUC_n frequency and frequency tolerance are locked to the local clock's frequency and frequency tolerance. The local clock frequency tolerance for the OTN is specified to be ± 20 ppm.

ODU maintenance signals (ODU AIS, OCI, LCK) are generated using a local clock. In a number of cases this local clock may be the clock of a server ODU signal over which the ODU signal is transported between equipment or through equipment (in one or more of the tributary slots). For these cases, the nominal justification ratio should be deployed to comply with the ODU's bit-rate tolerance specification.

12.2.1 ODU0, ODU1, ODU2, ODU3, ODU4, ODUC_n

The local clocks used to create the ODU0, ODU1, ODU2, ODU3, ODU4 and ODUC_n signals are generated by clock crystals that are also used for the generation of SDH STM-N signals. The bit rates of these ODU_k ($k=0,1,2,3,4$) and ODUC_n signals are therefore related to the STM-N bit rates and the bit-rate tolerances are the bit-rate tolerances of the STM-N signals.

The ODU0 bit rate is 50% of the STM-16 bit rate.

The ODU1 bit rate is 239/238 times the STM-16 bit rate.

The ODU2 bit rate is 239/237 times 4 times the STM-16 bit rate.

The ODU3 bit rate is 239/236 times 16 times the STM-16 bit rate.

The ODU4 bit rate is 239/227 times 40 times the STM-16 bit rate.

The ODUC_n bit rate is n times 239/226 times 40 times the STM-16 bit rate.

ODU1, ODU2 and ODU3 signals which carry an STM-N ($N = 16, 64, 256$) signal may also be generated using the timing of these client signals.

Refer to Table 7-2 for the nominal bit rates and bit-rate tolerances.

12.2.2 ODU2e

An ODU2e signal is generated using the timing of its client signal.

The ODU2e bit rate is 239/237 times the 10GBASE-R client bit rate.

Refer to Table 7-2 for the nominal bit rate and bit-rate tolerances.

12.2.3 ODUflex for CBR client signals

An ODUflex(CBR) signal is generated using the timing of its client signal.

The ODUflex bit rate is 239/238 times the adapted CBR client bit rate.

The client signal may have a bit-rate tolerance up to ± 100 ppm.

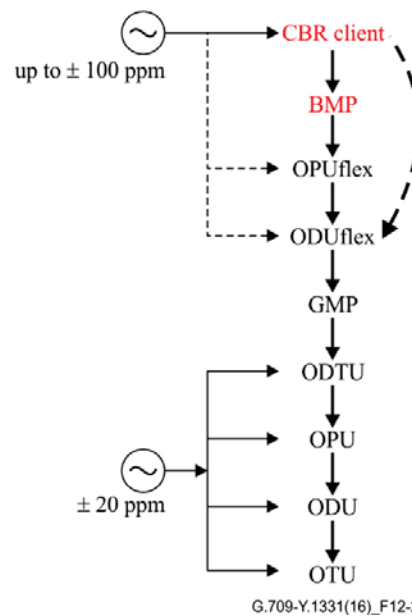


Figure 12-2 – ODUflex clock generation for CBR signals

12.2.4 ODUflex for PRBS and Null test signals

ODUflex(CBR) connections may be tested using a PRBS or NULL test signal as the client signal instead of the CBR client signal. For such a case, the ODUflex(PRBS) or ODUflex(NULL) signal should be generated with a frequency within the tolerance range of the ODUflex(CBR) signal.

If the CBR client clock is present such ODUflex(PRBS) or ODUflex(NULL) signal may be generated using the CBR client clock, otherwise the ODUflex(PRBS) or ODUflex(NULL) signal is generated using a local clock.

12.2.5 ODUflex for GFP-F mapped packet client signals

ODUflex(GFP) signals are generated using a local clock. This clock may be the local server ODUk (or OTUk) clock, the local ODUCn (or OTUCn) clock, or an equipment internal clock of the signal over which the ODUflex is carried through the equipment.

Any bit rate is possible for an ODUflex(GFP) signal, however it is suggested for maximum efficiency that the ODUflex(GFP) fills an integral number of tributary slots of the smallest server ODUk or ODUCn path over which the ODUflex(GFP) may be carried. The recommended bit-rates to meet this criterion are specified in Table 7-8. The derivation of the specific values is provided in Appendix XI.

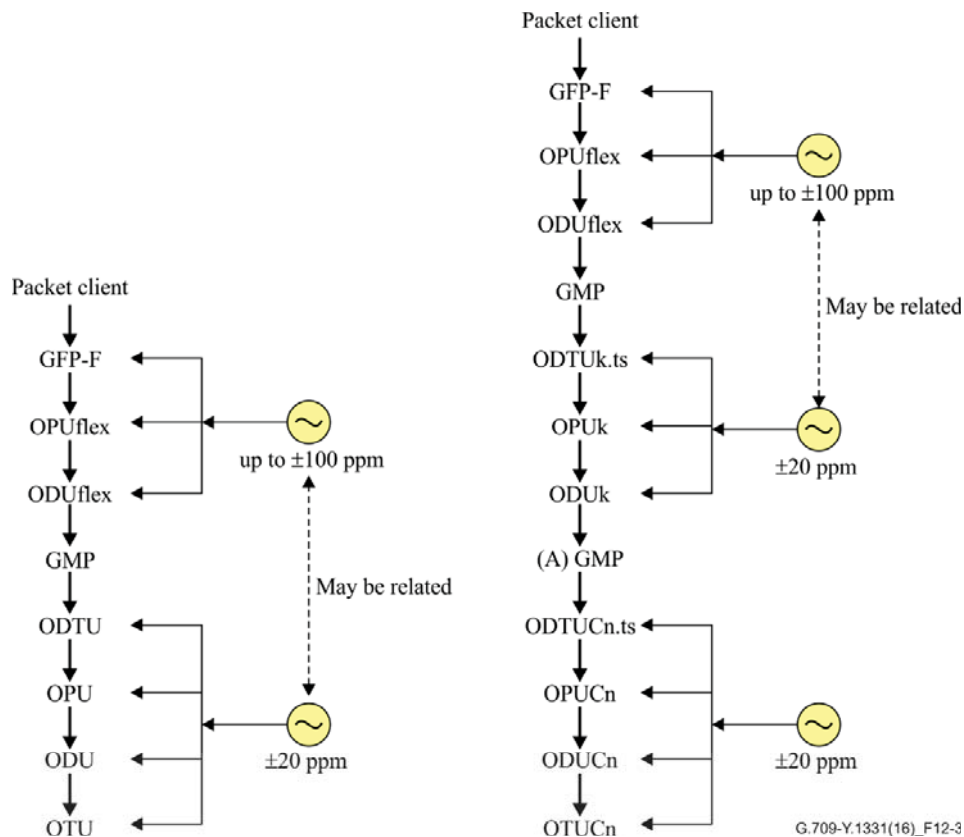


Figure 12-3 – ODUflex clock generation for GFP-F mapped packet client signals

12.2.6 ODUflex for IMP mapped client signals

ODUflex(IMP) signals are generated using a local clock or the timing of its client signal. The local clock may be the local server ODUk (or OTUk) clock, the local ODUCn (or OTUCn) clock, or an equipment internal clock of the signal over which the ODUflex is carried through the equipment.

In the first method, the clock of the server ODU is used as reference clock and adjusted by means of a fixed C_m value, similar to the case of ODUflex(GFP). The main difference with the ODUflex(GFP) clock rate is that this latter rate is the maximum rate that can be supported by M trib slots whereas the ODUflex(IMP) clock rate should not be such maximum rate, but instead just a rate that is sufficient to carry the FlexE client signal or the PKT client encapsulated in a FlexE client signal.

In the second method, a local free run clock is used as reference clock and adjusted by means of a fixed X/Y value.

In the third method, the clock of the FlexE Client signal is used as reference clock and adjusted by means of a fixed X/Y value, with $X = 239$ and $Y = 238$.

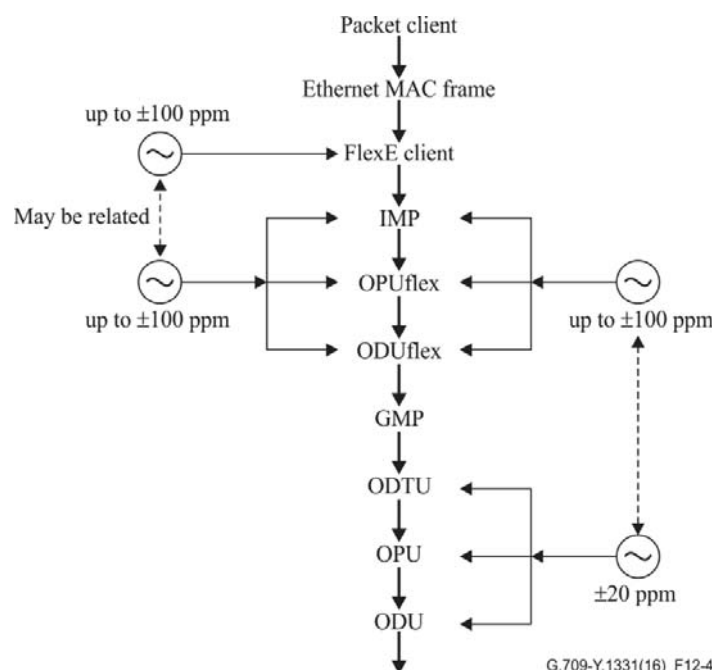
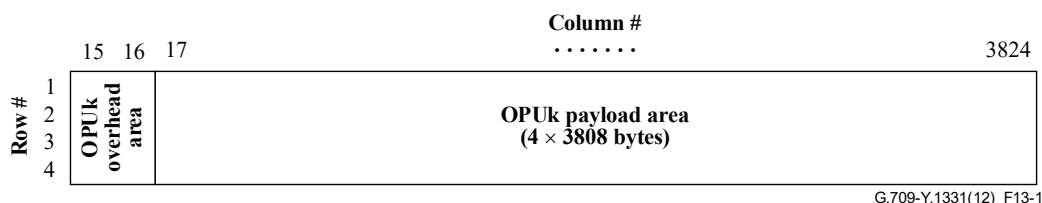


Figure 12-4 – ODUflex clock generation for IMP mapped packet client signals

13 Optical payload unit (OPU)

The OPU frame structure is shown in Figure 13-1. It is organized in an octet-based block frame structure with four rows and 3810 columns.

The OPU_k (k=0,1,2,2e,3,4,flex) frame structure contains one instance of the OPU frame structure. The OPU_{Cn} frame structure contains n frame and multi-frame synchronous instances of the OPU frame structure, numbered 1 to n (OPU #1 to OPU #n).



G.709-Y.1331(12)_F13-1

Figure 13-1 – OPU frame structure

The two main areas of the OPU frame are:

- OPU overhead area;
- OPU payload area.

Columns 15 to 16 of the OPU are dedicated to an OPU overhead area.

Columns 17 to 3824 of the OPU are dedicated to an OPU payload area.

NOTE – OPU column numbers are derived from the OPU columns in the ODU frame.

14 Overhead information carried over the OSC and OCC

The overhead information carried over the OSC consists of the OTS-O, OMS-O, OCh-O and OTSiG-O. The information content of this overhead is defined in clause 15. The format, structure and bit rate of this overhead is not defined in this Recommendation.

The overhead information carried over the OCC consists of the OCh-O. The information content of this overhead is defined in clause 15. The format, structure and bit rate of this overhead is defined in [ITU-T G.7712].

General management communications (COMMS)

Depending on an operator's logical management overlay network design, general management communications (COMMS) may also be transported within the OSC. Therefore, the OSC for some applications may also transport general management communications. General management communications may include signalling, voice/voiceband communications, software download, operator-specific communications, etc.

OTN synchronisation message channel (OSMC)

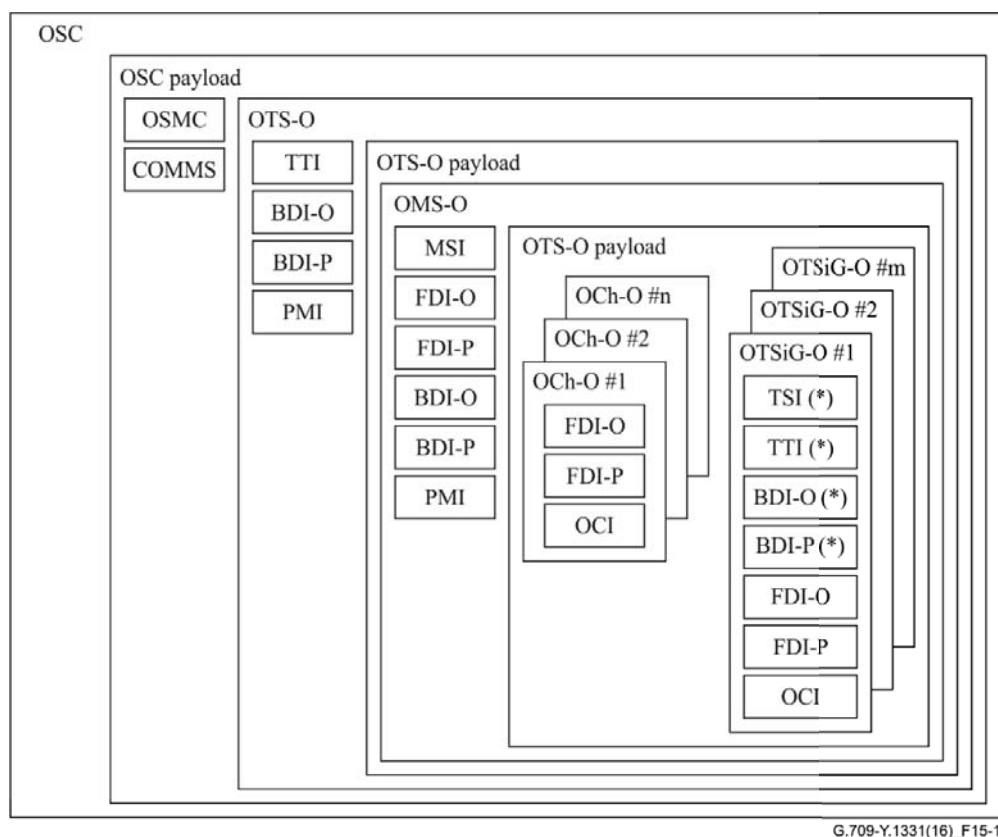
For synchronisation purposes, the OSC OSMC signal is defined as an OTN synchronisation message channel to transport SSM and PTP messages.

NOTE 1 – Support of OSC OSMC in a MOTUm interface is optional.

NOTE 2 – Equipment designed prior to Edition 4.6 of this recommendation may not be able to support OSC OSMC.

15 Overhead description

An overview of OTS-O, OMS-O, OCh-O and OTSiG-O information carried within the OSC is presented in Figure 15-1. An overview of OCh-O information carried within the OCC is presented in Figure 15-2.



(*) This overhead may be carried over the OSC or over a communication channel modulated on one or more OTSi within the OTSiG.

Figure 15-1 – OTS-O, OMS-O, OCh-O and OTSiG-O information carried within the OSC

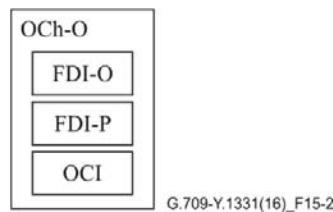


Figure 15-2 – OCh-O information carried within the OCC

An overview of OTUk, ODUk and OPUk overhead is presented in Figures 15-3 and 15-5. An overview of OTUCn, ODUCn and OPUCn overhead is presented in Figures 15-4 and 15-6.

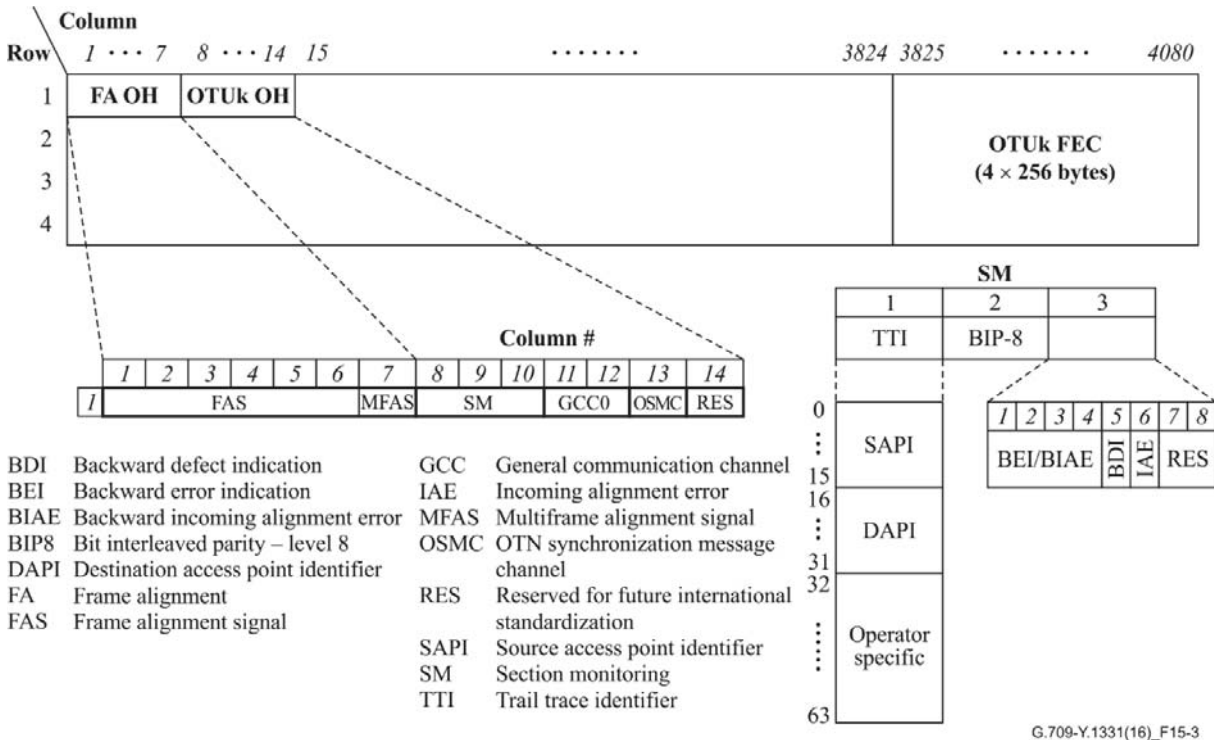


Figure 15-3 – OTUk frame structure, frame alignment and OTUk overhead

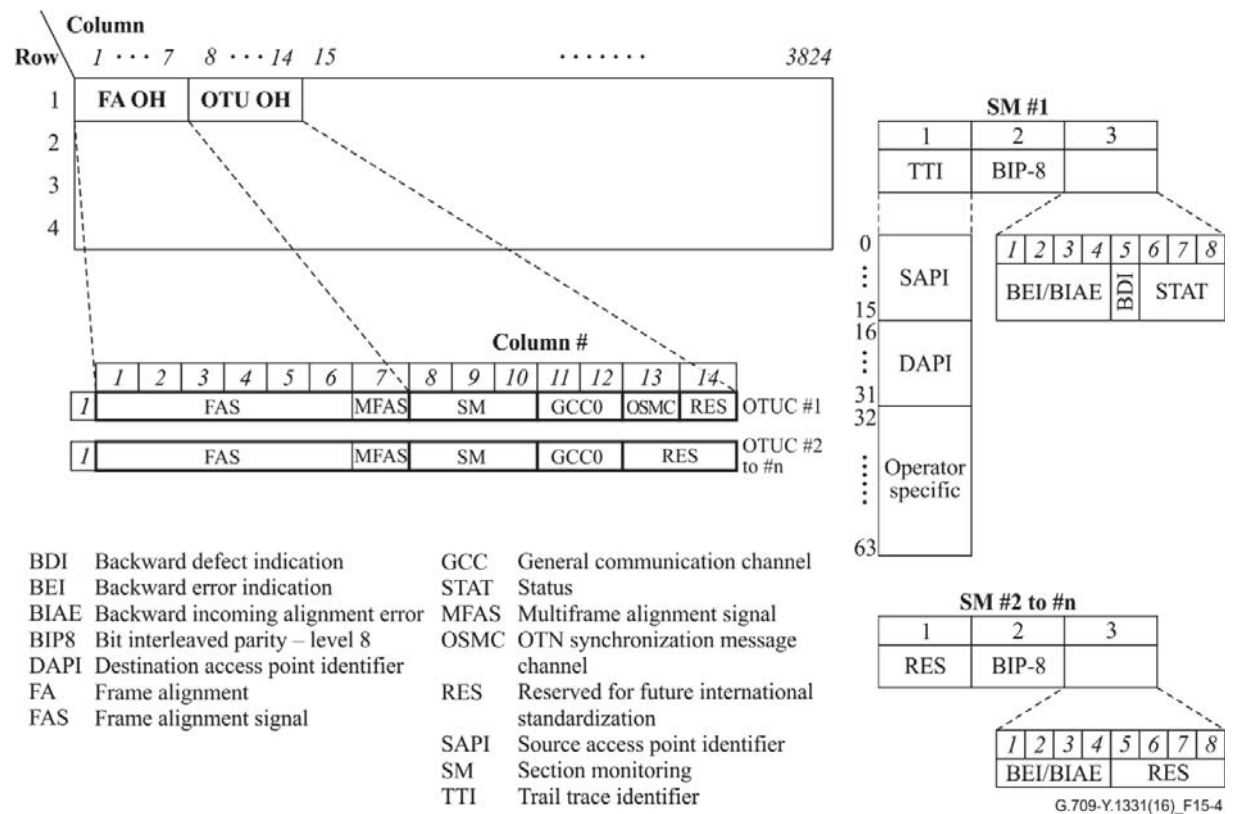


Figure 15-4 – OTUCn frame structure, frame alignment and OTUCn overhead

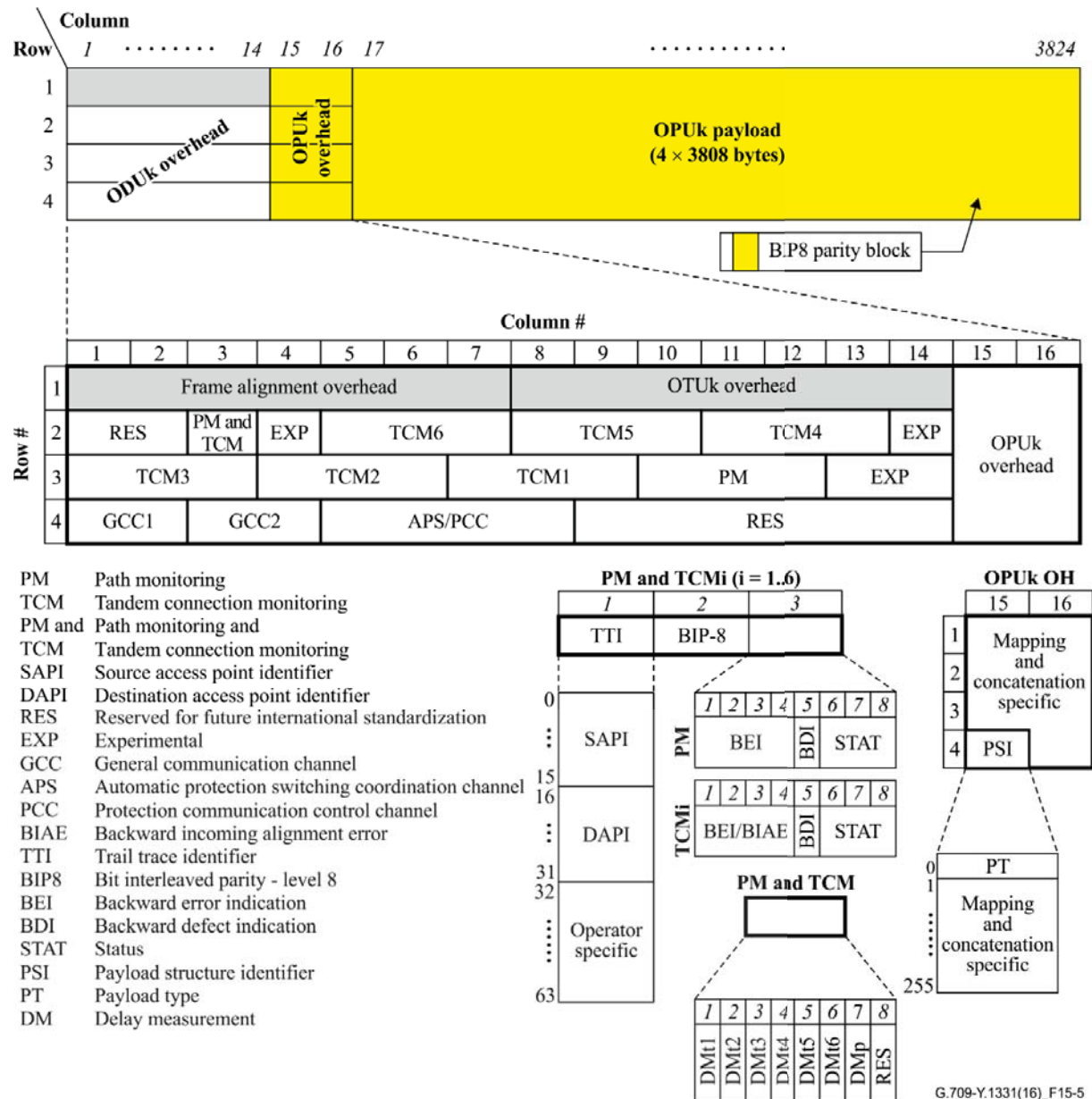
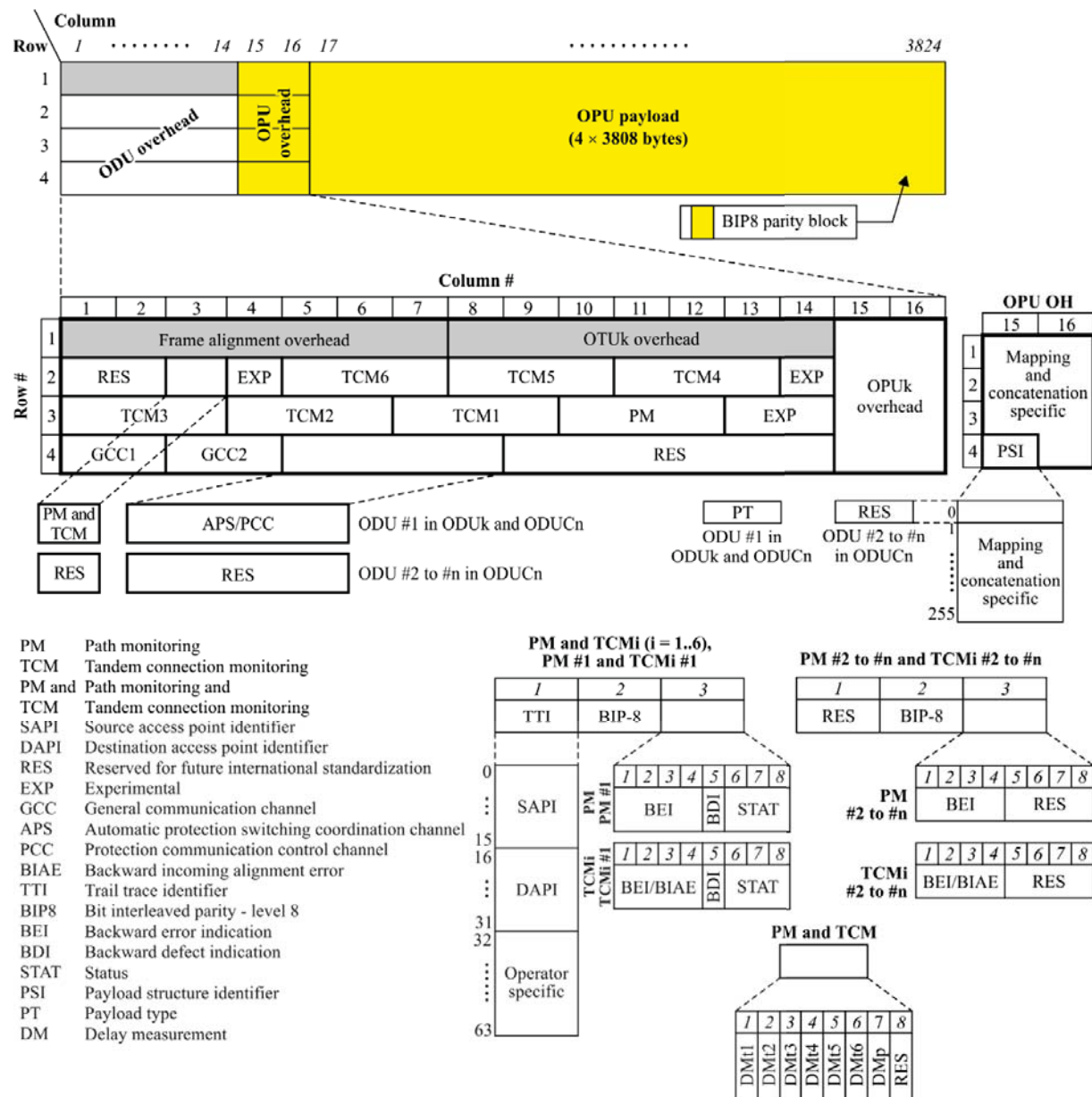


Figure 15-5 – ODUk frame structure, ODUk and OPUk overhead



G.709-Y.1331(16)_F15-6

Figure 15-6 – ODU_{Cn} frame structure, ODU_{Cn} and OPU_{Cn} overhead

15.1 Types of overhead

15.1.1 Optical payload unit overhead (OPU OH)

OPU OH information is added to the OPU information payload to create an OPU. It includes information to support the adaptation of client signals. The OPU OH is terminated where the OPU is assembled and disassembled. The specific OH format and coding is defined in clause 15.9.

15.1.2 Optical data unit overhead (ODU OH)

ODU OH information is added to the ODU information payload to create an ODU. It includes information for maintenance and operational functions to support ODU connections. The ODU OH consists of portions dedicated to the end-to-end ODU path and to six levels of tandem connection monitoring. The ODU path OH is terminated where the ODU is assembled and disassembled. The TC OH is added and terminated at the source and sink of the corresponding tandem connections, respectively. The specific OH format and coding is defined in clauses 15.6 and 15.8.

15.1.3 Optical transport unit overhead (OTU OH)

OTU OH information is part of the OTU signal structure. It includes information for operational functions to support the transport via one or more OCh connections. The OTU OH is terminated where the OTU signal is assembled and disassembled. The specific OH format and coding is defined in clauses 15.6 and 15.7.

The specific frame structure and coding for the non-standard OTUkV OH is outside the scope of this Recommendation. Only the required basic functionality that has to be supported is defined in clause 15.7.3.

15.1.4 OCh-O

OCh-O is the non-associated overhead information that accompanies an OCh-P. It includes information for maintenance functions to support fault management. The OCh-O is generated or modified at intermediate points along the OCh trail. All information elements of the OCh-O are terminated where the OCh-P signal is terminated.

The OCh-O information is defined in clause 15.5.

15.1.5 OMS-O

OMS-O is the non-associated overhead information that accompanies an OMS-P. It includes information for maintenance and operational functions to support optical multiplex sections. The OMS-O is generated or modified at intermediate points along the OMS trail and generated and terminated where the OMS-P signal is assembled and disassembled.

The OMS-O information is defined in clause 15.4.

15.1.6 OTS-O

OTS-O is the non-associated overhead information that accompanies an OTS-P. It includes information for maintenance and operational functions to support optical transmission sections. The OTS-O is terminated where the OTS-P signal is assembled and disassembled.

The OTS-O information is defined in clause 15.3.

15.1.7 General management communications overhead (COMMS OH)

COMMS OH information is added to the information payload to create a MOTUm interface signal. It provides general management communication between network elements. The specific frame structure and coding for the COMMS OH is outside the scope of this Recommendation.

15.1.8 OTSiG-O

OTSiG-O is the non-associated overhead information that accompanies an OTSiG. It includes information for maintenance functions to support fault management. The OTSiG-O TSI, TTI, BDI-P and BDI-O are generated where the OTSiA signal is assembled and the OCI, FDI-P and FDI-O are modified or generated at intermediate points along the OTSiA trail. All information elements of the OTSiG-O are terminated where the OTSiA is disassembled.

The OTSiG-O information is defined in clause 15.5.

15.2 Trail trace identifier and access point identifier definition

A trail trace identifier (TTI) is defined as a 64-byte string with the following structure (see Figure 15-7):

- TTI[0] contains the SAPI[0] character, which is fixed to all-0s.
- TTI[1] to TTI[15] contain the 15-character source access point identifier (SAPI[1] to SAPI[15]).

- TTI[16] contains the DAPI[0] character, which is fixed to all-0s.
- TTI[17] to TTI[31] contain the 15-character destination access point identifier (DAPI[1] to DAPI[15]).
- TTI[32] to TTI[63] are operator specific.

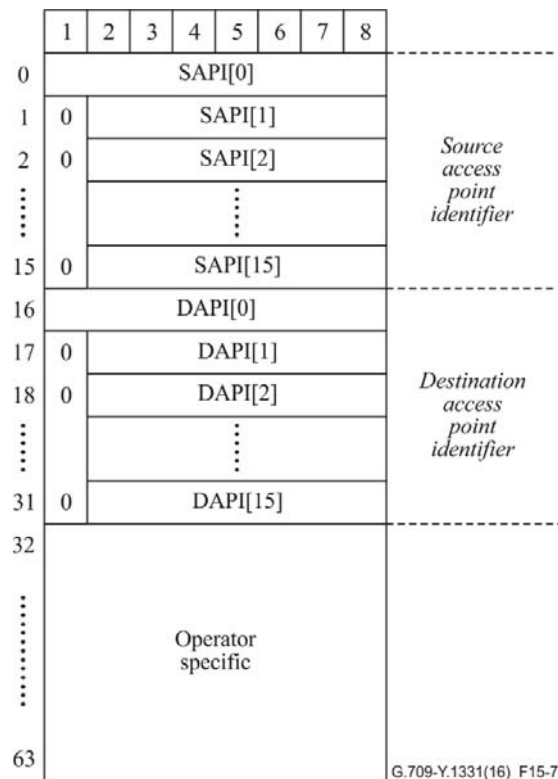


Figure 15-7 – TTI structure

The features of access point identifiers (APIs) are:

- Each access point identifier must be globally unique in its layer network.
- Where it may be expected that the access point may be required for path set-up across an inter-operator boundary, the access point identifier must be available to other network operators.
- The access point identifier should not change while the access point remains in existence.
- The access point identifier should be able to identify the country and network operator which is responsible for routing to and from the access point.
- The set of all access point identifiers belonging to a single administrative layer network should form a single access point identification scheme.
- The scheme of access point identifiers for each administrative layer network can be independent from the scheme in any other administrative layer network.

It is recommended that the ODUk, OTUk and OTS should each have the access point identification scheme based on a tree-like format to aid routing control search algorithms. The access point identifier should be globally unambiguous.

The access point identifier (SAPI, DAPI) shall consist of a three-character international segment and a twelve-character national segment (NS) (see Figure 15-8). These characters shall be coded according to [ITU-T T.50] (International Reference Alphabet – 7-bit coded character set for information exchange).

IS character #			NS character #											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CC			ICC	UAPC										
CC			ICC			UAPC								
CC			ICC				UAPC							
CC			ICC					UAPC						
CC			ICC						UAPC					
CC			ICC							UAPC				
CC			ICC								UAPC			

Figure 15-8 – Access point identifier structure

The international segment field provides a three-character ISO 3166 geographic/political country code (G/PCC). The country code shall be based on the three-character uppercase alphabetic ISO 3166 country code (e.g., USA, FRA).

The national segment field consists of two subfields: the ITU carrier code (ICC) followed by a unique access point code (UAPC).

The ITU carrier code is a code assigned to a network operator/service provider, maintained by the ITU-T Telecommunication Standardization Bureau (TSB) as per [ITU-T M.1400]. This code shall consist of 1-6 left-justified characters, alphabetic, or leading alphabetic with trailing numeric.

The unique access point code shall be a matter for the organization to which the country code and ITU carrier code have been assigned, provided that uniqueness is guaranteed. This code shall consist of 6-11 characters, with trailing NUL, completing the 12-character national segment.

15.3 OTS-O description

The following OTS-O information elements are defined:

- OTS-TTI
- OTS-BDI-P
- OTS-BDI-O
- OTS-PMI

15.3.1 OTS trail trace identifier (TTI)

The OTS-TTI is defined to transport a 64-byte TTI as specified in clause 15.2 for OTSn section monitoring.

15.3.2 OTS backward defect indication – Payload (BDI-P)

For OTS section monitoring, the OTS-BDI-P signal is defined to convey in the upstream direction the OTS-P signal fail status detected in the OTS-P termination sink function.

15.3.3 OTS backward defect indication – Overhead (BDI-O)

For OTS section monitoring, the OTS-BDI-O signal is defined to convey in the upstream direction the OTS-O signal fail status detected in the OTS-O termination sink function.

15.3.4 OTS payload missing indication (PMI)

The OTS PMI is a signal sent downstream as an indication that upstream at the source point of the OTS-P signal no payload is added, in order to suppress the report of the consequential loss of signal condition.

15.4 OMS-O description

The following OMS-O information elements are defined:

- OMS-FDI-P
- OMS-FDI-O
- OMS-BDI-P
- OMS-BDI-O
- OMS-PMI
- OMS-MSI

15.4.1 OMS forward defect indication – Payload (FDI-P)

For OMS section monitoring, the OMSn-FDI-P signal is defined to convey in the downstream direction the OMS-P signal status (normal or failed).

15.4.2 OMS forward defect indication – Overhead (FDI-O)

For OMS section monitoring, the OMSn-FDI-O signal is defined to convey in the downstream direction the OMS-O signal status (normal or failed).

15.4.3 OMS backward defect indication – Payload (BDI-P)

For OMS section monitoring, the OMSn-BDI-P signal is defined to convey in the upstream direction the OMS-P signal fail status detected in the OMS-P termination sink function.

15.4.4 OMS backward defect indication – Overhead (BDI-O)

For OMS section monitoring, the OMSn-BDI-O signal is defined to convey in the upstream direction the OMS-O signal fail status detected in the OMS-O termination sink function.

15.4.5 OMS payload missing indication (PMI)

The OMS PMI is a signal sent downstream as an indication that upstream at the source point of the OMS-P signal none of the frequency slots contain an optical tributary signal, in order to suppress the report of the consequential loss of signal condition.

15.4.6 OMS multiplex structure identifier (MSI)

The OMS multiplex structure identifier (MSI) signal encodes the OCh-P/OTSiG multiplex structure and occupied frequency slots in the OMS-P at the source end point. It is sent downstream to enable detection of OCh-P/OTSiG multiplex structure configuration mismatches between sink and source end points.

The OMS multiplex structure identifier (MSI) signal also encodes the media channel structure which is independent of the OCh-P/OTSiG in the OMS-P at the source and sink end points. It is sent downstream to enable detection of media channel structure configuration mismatches between sink and source end points.

The OMS MSI is defined for flex grid MOTUm interfaces. Fixed grid MOTUm interfaces may not support the OMS MSI. Flex grid MOTUm interfaces designed prior to Edition 5.0 of this Recommendation may not support the OMS MSI.

15.5 OCh-O and OTSiG-O description

The following OCh-O and OTSiG-O information elements are defined:

- OCh-FDI-P and OTSiA-FDI-P
- OCh-FDI-O and OTSiA-FDI-O

- OCh-OCI and OTSiA-OCI
- OTSiA-BDI-P
- OTSiA-BDI-O
- OTSiA-TTI
- OTSiG-TSI

15.5.1 OCh and OTSiA forward defect indication – Payload (FDI-P)

For OCh and OTSiA trail monitoring, the OCh-FDI-P or OTSiA-FDI-P signal is defined to convey in the downstream direction the OCh-P or OTSiG signal status (normal or failed).

15.5.2 OCh and OTSiA forward defect indication – Overhead (FDI-O)

For OCh and OTSiA trail monitoring, the OCh-FDI-O or OTSiA-FDI-O signal is defined to convey in the downstream direction the OCh-O or OTSiG-O overhead signal status (normal or failed).

15.5.3 OCh and OTSiA open connection indication (OCI)

The OCh and OTSiA OCI is a signal sent downstream as an indication that upstream in a connection function the OCh (i.e., OCh-P and OCh-O) or OTSiA (i.e., OTSiG and OTSiG-O) matrix connection is opened as a result of a management command. The consequential detection of the OCh or OTSiA loss of signal condition at the OCh or OTSiA termination point can now be related to an open matrix.

15.5.4 OCh-O transport over SOTUm interface

In the case of a MOTUm interface, the coding and method of OCh-O information transfer is vendor specific. For a SOTUm interface, the OCh-O is transferred over the overhead communication network (OCN) as described in [ITU-T G.7712].

NOTE – A SOTUm interface which transfers OCh-O over an OCN cannot provide fate-sharing of the OCh-O with the OCh-P across this interface.

The OCh FDI-P, FDI-O and OCI Overhead primitives are communicated over the OCN. The specification of the encapsulation, identification and transmission of this information is outside the scope of this Recommendation and specified in [ITU-T G.7712]. This information must be communicated with the peers such that the OCh-O primitives come into sync within one second in the absence of changes to OCh FDI-P, FDI-O or OCI. In the event of changes to any OCh-O primitive, the update must be sent within 10 ms of the change and with a mechanism to guarantee receipt in the event of packet loss.

15.5.5 OTSiA backward defect indication – Payload (BDI-P)

For OTSiA path monitoring, the OTSiA-BDI-P signal is defined to convey in the upstream direction the OTSiG signal fail status detected in the OTSiG termination sink function.

15.5.6 OTSiA backward defect indication – Overhead (BDI-O)

For OTSiA path monitoring, the OTSiA-BDI-O signal is defined to convey in the upstream direction the OTSiG-O signal fail status detected in the OTSiG-O termination sink function.

15.5.7 OTSiA trail trace identifier (TTI)

The OTSiA-TTI is defined to transport a 64-byte TTI as specified in clause 15.2 for OTSiA path monitoring.

15.5.8 OTSiG transmitter structure identifier (TSI)

The OTSiG-TSI is defined to transport a TSI for monitoring consistent configuration of the OTSiG transmitter(s) at source end points and OTSiG receiver(s) at sink end points.

15.6 OTU/ODU frame alignment OH description

15.6.1 OTU/ODU frame alignment overhead location

The OTU/ODU frame alignment overhead location is shown in Figure 15-9. The OTU/ODU frame alignment overhead is applicable for both the OTU and ODU signals.

The OTUk/ODUk contains one instance of OTU/ODU frame alignment overhead. The OTUCn/ODUCn contains n instances of OTU/ODU frame alignment overhead, numbered 1 to n.

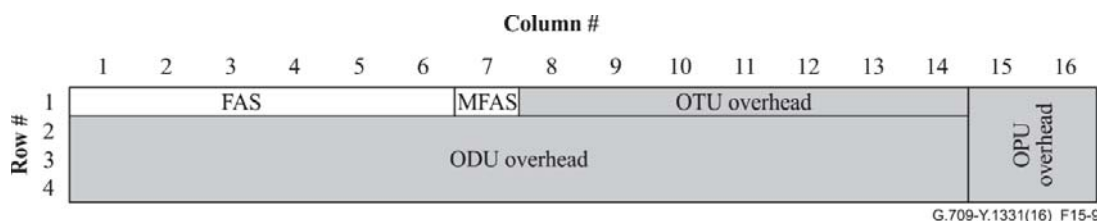


Figure 15-9 – OTU/ODU frame alignment overhead

15.6.2 OTU/ODU frame alignment overhead definition

15.6.2.1 Frame alignment signal (FAS)

A six byte OTU-FAS signal (see Figure 15-10) is defined in row 1, columns 1 to 6 of the OTU overhead. OA1 is "1111 0110". OA2 is "0010 1000".

The OTUk contains one instance of OTU multi-frame alignment overhead. The OTUCn contains n instances of OTU multi-frame alignment overhead, numbered 1 to n.

FAS OH byte 1								FAS OH byte 2								FAS OH byte 3								FAS OH byte 4								FAS OH byte 5								FAS OH byte 6							
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
OA1								OA1								OA1								OA2								OA2								OA2							

G.709-Y.1331(16)_F15-10

Figure 15-10 – Frame alignment signal overhead structure

15.6.2.2 Multiframe alignment signal (MFAS)

Some of the OTU and ODU overhead signals will span multiple OTU/ODU frames. Examples are the TTI and TCM-ACT overhead signals. These and other multiframe structured overhead signals require multiframe alignment processing to be performed, in addition to the OTU/ODU frame alignment.

A single multiframe alignment signal (MFAS) byte is defined in row 1, column 7 of the OTU/ODU overhead for this purpose (see Figure 15-11). The value of the MFAS byte will be incremented each OTU/ODU frame and provides as such a 256-frame multiframe.

The OTUk contains one instance of OTU multi-frame alignment overhead. The OTUCn contains n instances of OTU multi-frame alignment overhead, numbered 1 to n. All n MFAS bytes carry the same 256-frame sequence and in each frame all n MFAS bytes carry the same value.



Figure 15-11 – Multiframe alignment signal overhead

Individual OTU/ODU overhead signals may use this central multiframe to lock their 2-frame, 4-frame, 8-frame, 16-frame, 32-frame, etc., multiframe to the principal frame.

NOTE 1 – The 80-frame OPU4 multiframe cannot be supported. A dedicated 80-frame OPU4 multiframe indicator (OMFI) is used instead.

NOTE 2 – The 20-frame OPUCn multiframe cannot be supported by MFAS. A dedicated 20-frame OPUCn multiframe indicator (OMFI) is used instead.

15.7 OTU OH description

15.7.1 OTU overhead location

The OTU overhead location is shown in Figures 15-12 and 15-13.

The OTUk contains one instance of OTU overhead. The OTUCn contains n instances of OTU overhead, numbered 1 to n (OTU OH #1 to OTU OH #n).

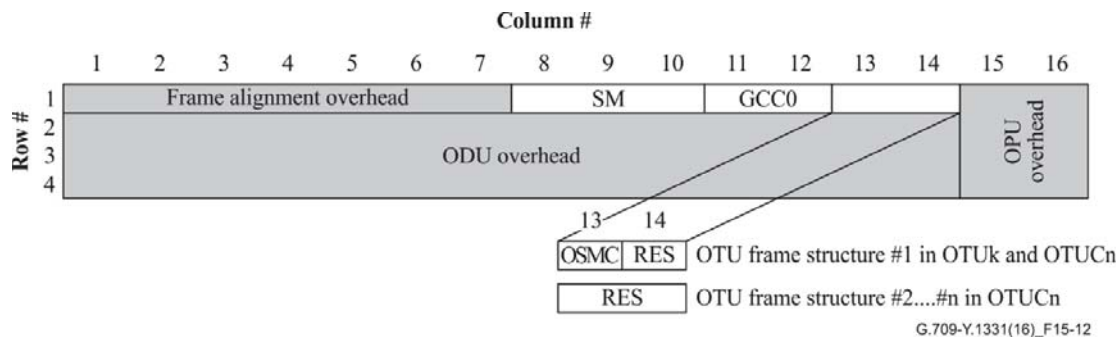


Figure 15-12 – OTU overhead

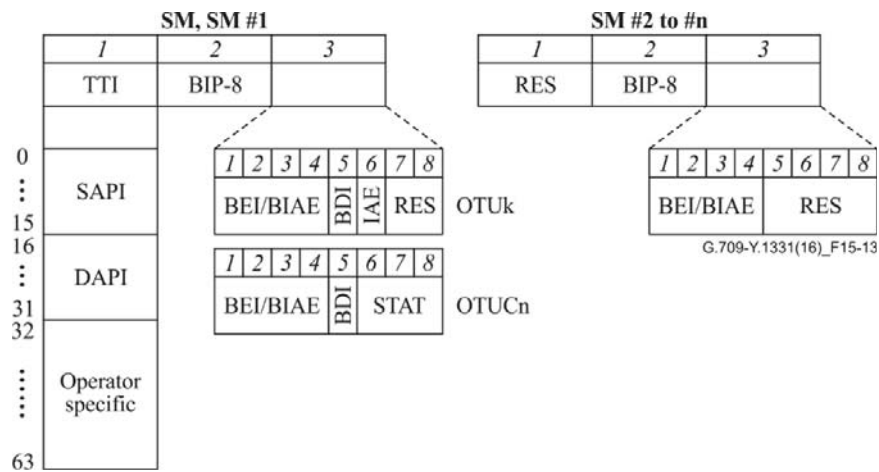


Figure 15-13 – OTU section monitoring overhead

15.7.2 OTU overhead definition

15.7.2.1 OTU section monitoring (SM) overhead

One field of OTU section monitoring (SM) overhead is defined in row 1, columns 8 to 10 to support section monitoring.

The OTU_k contains one instance of OTU SM overhead. The OTU_{Cn} contains n instances of the OTU SM overhead, numbered 1 to n (SM #1 to SM #n).

The SM and SM #1 field contains the following subfields (see Figure 15-13):

- trail trace identifier (TTI);
- bit interleaved parity (BIP-8);
- backward defect indication (BDI);
- backward error indication and backward incoming alignment error (BEI/BIAE);
- incoming alignment error (IAE);
- status bits indicating the presence of an incoming alignment error or a maintenance signal (STAT);
- bits reserved for future international standardization (RES).

The SM #2 to #n fields contain the following subfields (see Figure 15-13):

- bit interleaved parity (BIP-8);
- backward error indication and backward incoming alignment error (BEI/BIAE);
- bits reserved for future international standardization (RES).

15.7.2.1.1 OTU SM trail trace identifier (TTI)

For section monitoring, a one-byte trail trace identifier (TTI) overhead is defined to transport the 64-byte TTI signal specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1].

The OTU_k and OTU_{Cn} contain one instance of OTU TTI overhead.

The 64-byte TTI signal shall be aligned with the OTU multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at OTU multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.7.2.1.2 OTU SM error detection code (BIP-8)

For section monitoring, a one-byte error detection code signal is defined in the OTU SM overhead. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation *BIP-8* refers only to the number of BIP bits and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8 refer to BIP-X definition in [ITU-T G.707].

The OTU BIP-8 is computed over the bits in the OPU (columns 15 to 3824) area of OTU frame *i*, and inserted in the OTU BIP-8 overhead location in OTU frame *i*+2 (see Figure 15-14).

The OTU_k contains one instance of OTU BIP-8 overhead. The OTU_{Cn} contains *n* instances of the OTU BIP-8 overhead, numbered 1 to *n* (BIP-8 #1 to BIP-8 #*n*).

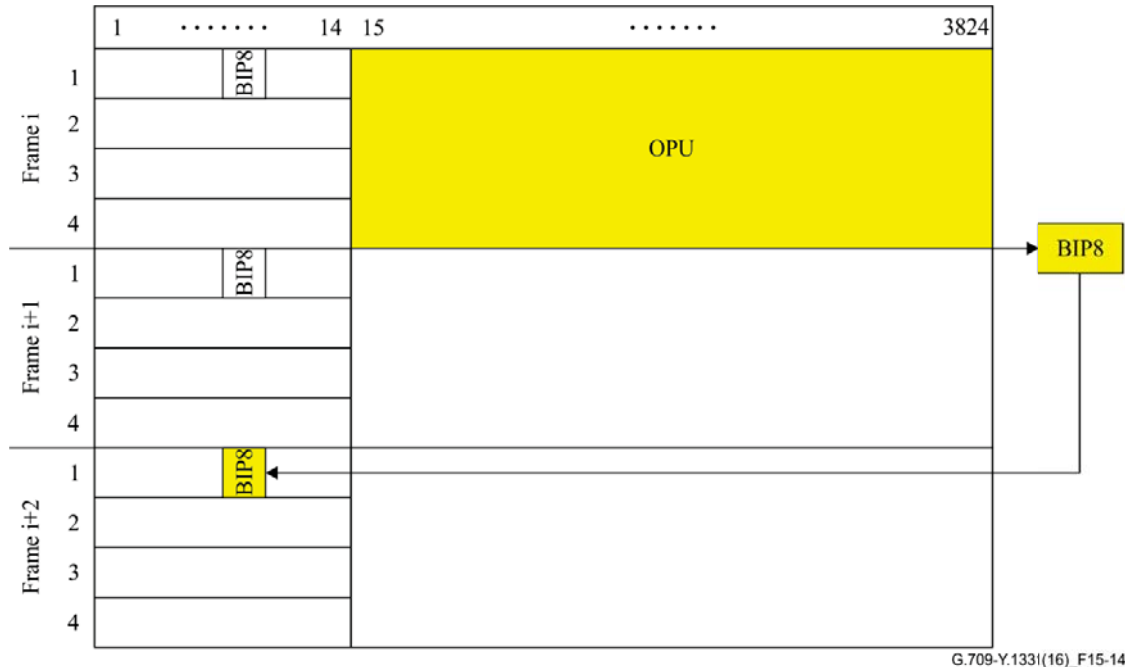


Figure 15-14 – OTU SM BIP-8 computation

15.7.2.1.3 OTU SM backward defect indication (BDI)

For section monitoring, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a section termination sink function in the upstream direction.

BDI is set to "1" to indicate an OTU backward defect indication; otherwise, it is set to "0".

The OTU_k and OTU_{Cn} contain one instance of OTU BDI overhead.

15.7.2.1.4 OTU SM backward error indication and backward incoming alignment error (BEI/BIAE)

For section monitoring, a four-bit backward error indication (BEI) and backward incoming alignment error (BIAE) signal is defined. This signal is used to convey in the upstream direction the count of interleaved-bit blocks that have been detected in error by the corresponding OTU section monitoring sink using the BIP-8 code. It is also used to convey in the upstream direction an incoming alignment error (IAE) condition that is detected in the corresponding OTU section monitoring sink in the IAE overhead.

During an IAE condition the code "1011" is inserted into the BEI/BIAE field and the error count is ignored. Otherwise the error count (0-8) is inserted into the BEI/BIAE field. The remaining six possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-1) and BIAE not active.

The OTU_k contains one instance of OTU BEI/BIAE overhead. The OTUC_n contains n instances of the OTU BEI/BIAE overhead, numbered 1 to n (BEI/BIAE #1 to BEI/BIAE #n).

NOTE – The BIAE indication of an OTUC_n is transported n times (ie. in OTUC SM # 1 to SM #n) and detected in OTUC SM #1 only.

Table 15-1 – OTU SM BEI/BIAE interpretation

OTU SM BEI/BIAE bits	1 2 3 4	BIAE	BIP violations
0 0 0 0		false	0
0 0 0 1		false	1
0 0 1 0		false	2
0 0 1 1		false	3
0 1 0 0		false	4
0 1 0 1		false	5
0 1 1 0		false	6
0 1 1 1		false	7
1 0 0 0		false	8
1 0 0 1, 1 0 1 0		false	0
1 0 1 1		true	0
1 1 0 0 to 1 1 1 1		false	0

15.7.2.1.5 OTU_k SM incoming alignment error overhead (IAE)

A single-bit incoming alignment error (IAE) signal is defined to allow the S-CMEP ingress point to inform its peer S-CMEP egress point that an alignment error in the incoming signal has been detected.

IAE is set to "1" to indicate a frame alignment error, otherwise it is set to "0".

The S-CMEP egress point may use this information to suppress the counting of bit errors, which may occur as a result of a frame phase change of the OTU_k at the ingress of the section.

15.7.2.1.6 OTU SM reserved overhead (RES)

For section monitoring of the OTU_k, two bits in the SM overhead are reserved (RES) for future international standardization. They are set to "00".

For section monitoring of the OTUC_n, 12 bits in the SM overhead in the OTU frame structures #2 to #n are reserved for future international standardization. The value of these bits is set to "0".

15.7.2.1.7 OTUC_n SM status (STAT)

For section monitoring, three bits are defined as status bits (STAT). They indicate the presence of a maintenance signal or if there is an incoming alignment error at the source S-CMEP, (see Table 15-2).

Table 15-2 – OTUCn SM status interpretation

bits	SM byte 3 6 7 8	Status
	0 0 0	Reserved for future international standardization
	0 0 1	In use without IAE
	0 1 0	In use with IAE
	0 1 1	Reserved for future international standardization
	1 0 0	Reserved for future international standardization
	1 0 1	Reserved for future international standardization
	1 1 0	Reserved for future international standardization
	1 1 1	Maintenance signal: OTUCn-AIS

A S-CMEP ingress point sets these bits to either "001" to indicate to its peer S-CMEP egress point that there is no incoming alignment error (IAE), or to "010" to indicate that there is an incoming alignment error.

The S-CMEP egress point may use this information to suppress the counting of bit errors, which may occur as a result of a frame phase change of the ODUCn at the ingress of the section.

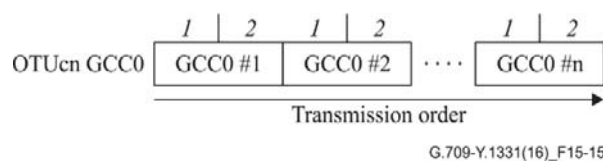
15.7.2.2 OTU general communication channel 0 (GCC0)

Two bytes are allocated in the OTU overhead to support a general communications channel or a discovery channel as specified in [ITU-T G.7714.1] between OTU termination points.

This general communication channel is a clear channel and any format specification is outside of the scope of this Recommendation. These bytes are located in row 1, columns 11 and 12 of the OTU overhead.

The OTUk contains one instance of OTU GCC0 overhead. The OTUCn contains n instances of the OTU GCC0 overhead, numbered 1 to n (GCC0 #1 to GCC0 #n).

The GCC0 #1 to #n overhead are combined to provide one communication channel as illustrated in Figure 15-15 with an approximated bandwidth of $n \times 13.768$ Mbit/s.

**Figure 15-15 – OTUCn GCC0 transmission order**

15.7.2.3 OTU reserved overhead (RES)

One byte of the OTU overhead in OTU frame structure #1 is reserved for future international standardization. This byte is located in row 1, column 14. This byte is set to all-0s.

Two bytes of the OTU overhead in OTU frame structures #2 to #n are reserved for future international standardization. These bytes are located in the OTU overhead in row 1, columns 13 and 14. These bytes are set to all-0s.

15.7.2.4 OTU OTN synchronisation message channel (OSMC)

For synchronisation purposes, one byte is defined in the OTU overhead as an OTN synchronisation message channel to transport SSM and PTP messages within SOTU and MOTU interfaces. The OSMC bandwidth is listed in Table 15-3.

The OTU_k and OTU_{Cn} contain one instance of OTU OSMC overhead.

NOTE 1 – OTU OSMC is not defined for MOTU_m and SOTU_m interfaces.

NOTE 2 – Support of OTU OSMC in a SOTU or MOTU interface is optional.

NOTE 3 – Equipment designed prior to Edition 5.0 of this Recommendation may not be able to support OTU OSMC via their SOTU or MOTU interfaces.

NOTE 4 – SOTU and MOTU interfaces with vendor specific application identifiers may support an OSMC function. The encapsulation of the messages and overhead location are then vendor specific.

Table 15-3 – OSMC bandwidth

OTU_k	OSMC bandwidth (kbit/s)
OTU1	163.361
OTU2	656.203
OTU3	2,635.932
OTU4	6,851.101
OTU _{Cn}	6,881.418

The SSM and PTP messages within a SOTU or MOTU interface are encapsulated into GFP-F frames as specified into [ITU-T G.7041]. PTP event messages are timestamped and after encapsulation into GFP-F frames inserted into the OSMC as specified in clause 15.7.2.4.1. GFP-F encapsulated SSM messages (and PTP non-event messages) are inserted into the OSMC at the earliest opportunity. GFP Idle frames may be inserted between successive GFP frames.

The mapping of generic framing procedure (GFP) frames is performed by aligning the byte structure of every GFP frame with the byte of the OSMC overhead field. Since the GFP frames are of variable length and longer than one byte, a frame crosses the OTU_k ($k=1,2,3,4$) and OTU_{Cn} frame boundary.

15.7.2.4.1 Generation of event message timestamps

15.7.2.4.1.1 SOTU and MOTU interface event message timestamp point

The SOTU and MOTU interface message timestamp point [ITU-T G.8260] for a PTP event message transported over the OSMC shall be the X-frame multiframe event preceding the beginning of the GFP frame in which the PTP event message is carried. See Figure 15-16. Since the GFP frames may be longer than X-4 bytes, a frame may cross the X-frame multiframe boundary. The X-frame multiframe contains frames numbered 0, 1, ..., X-1.

15.7.2.4.1.2 Event timestamp generation

All PTP event messages are timestamped on egress and ingress SOTU and MOTU interfaces. The timestamp shall be the time at which the event message timestamp point passes the reference plane [ITU-T G.8260] marking the boundary between the PTP node (i.e., OTN node) and the network.

OTU_k

Event message timestamps are generated every X-frame multiframe period at the OTU_k Access Point. X is 64 for $k = 1$ and 256 for $k = 2,3,4$.

The first byte of a GFP(PTP event message) frame is inserted into the OTU_k OSMC between 4 and X-1 frames after the X-frame multiframe boundary.

NOTE 1 – The first byte of a GFP(PTP event message) frame is not inserted into the OTU_k OSMC in frame 0...3 of the X-frame multiframe.

NOTE 2 – The guard band of 4 frames is defined to support the addition of PTP event messages on existing SOTU and MOTU interface ports.

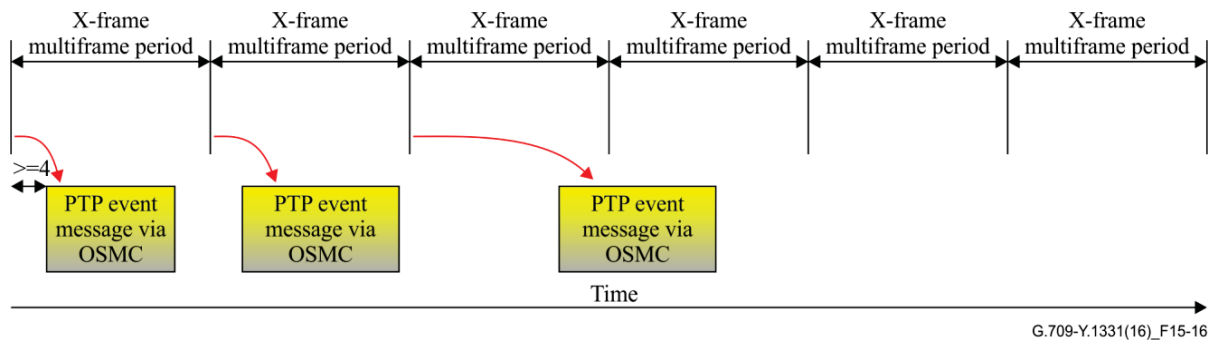


Figure 15-16 – Timing diagram example for OTUk

OTUCn

For further study.

NOTE 3 – This time synchronization over SOTU and MOTU interface implementation does not generate event message timestamps using a point other than the message timestamp point [ITU-T G.8260].

In this time synchronization over SOTU and MOTU interface implementation, the timestamps are generated at a point removed from the reference plane. Furthermore, the time offset from the reference plane is likely to be different for inbound and outbound event messages. To meet the requirement of this subclause, the generated timestamps should be corrected for these offsets. Figure 19 in [b-IEEE 1588] illustrates these offsets. Based on this model, the appropriate corrections are as follows:

$$\langle \text{egressTimestamp} \rangle = \langle \text{egressMeasuredTimestamp} \rangle + \text{egressLatency}$$

$$\langle \text{ingressTimestamp} \rangle = \langle \text{ingressMeasuredTimestamp} \rangle - \text{ingressLatency}$$

where the actual timestamps $\langle \text{egressTimestamp} \rangle$ and $\langle \text{ingressTimestamp} \rangle$ measured at the reference plane are computed from the detected, i.e., measured, timestamps by their respective latencies. Failure to make these corrections results in a time offset between the slave and master clocks.

15.7.3 OTUkV overhead

The functionally standardized OTUkV frame should support, as a minimum capability, section monitoring functionality comparable to the OTUk section monitoring (see clause 15.7.2.1) with a trail trace identifier as specified in clause 15.2. Further specification of this overhead is outside the scope of this Recommendation.

15.8 ODU OH description

15.8.1 ODU OH location

The ODU overhead location is shown in Figures 15-17, 15-18 and 15-19.

The ODUk contains one instance of ODU overhead. The ODUCn contains n instances of ODU overhead, numbered 1 to n (ODU OH #1 to ODU OH #n).

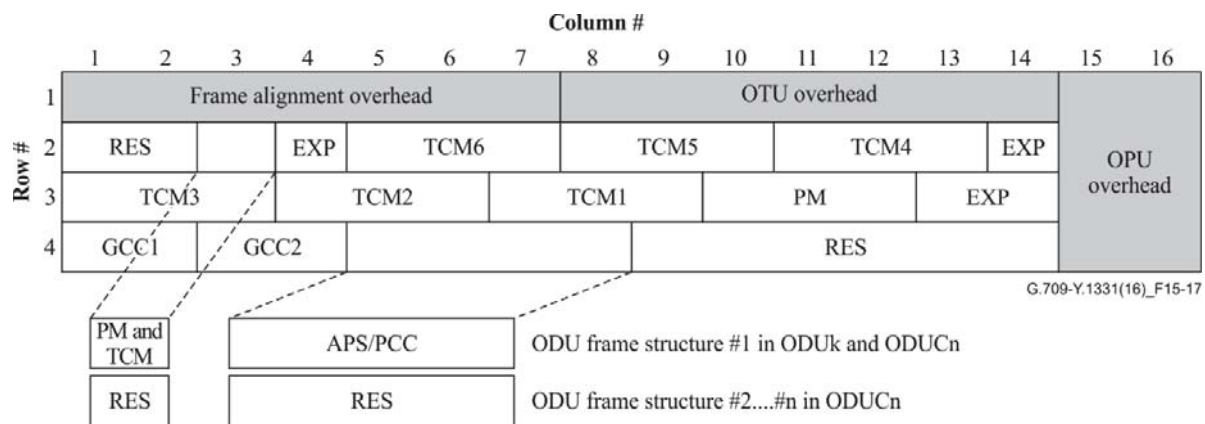


Figure 15-17 – ODU overhead

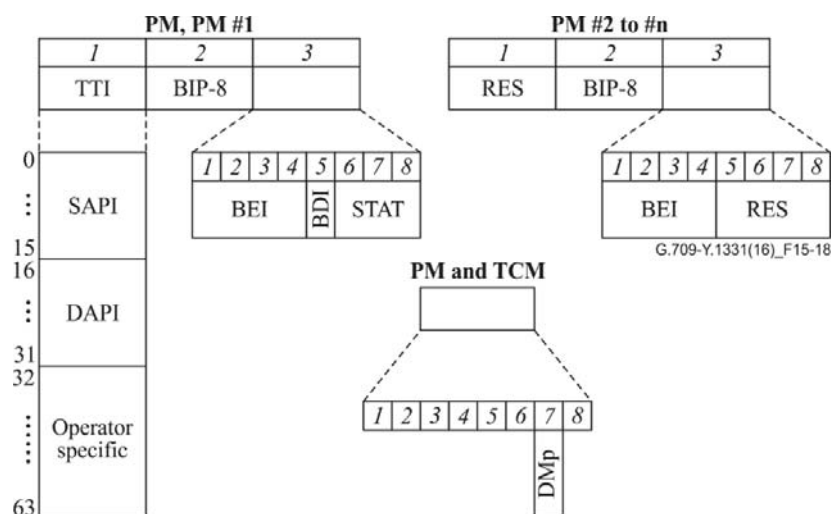


Figure 15-18 – ODU path monitoring overhead

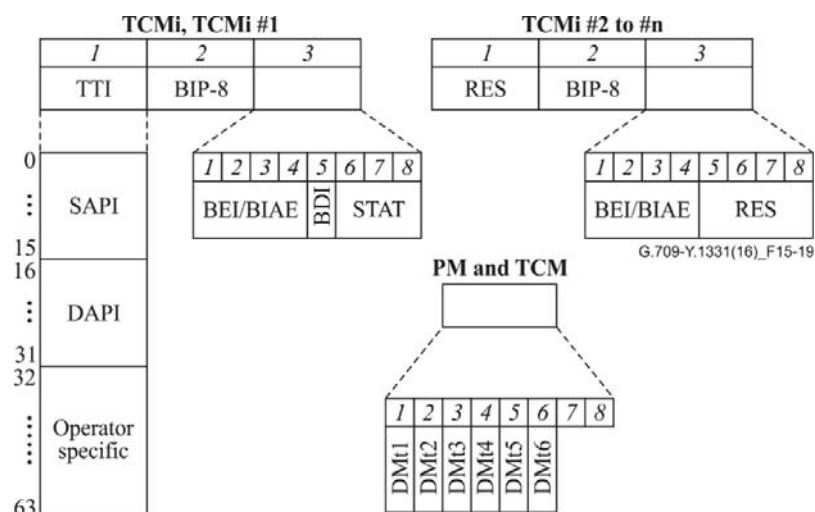


Figure 15-19 – ODU tandem connection monitoring #i overhead

15.8.2 ODU OH definition

15.8.2.1 ODU path monitoring (PM) overhead

One field of an ODU path monitoring overhead (PM) is defined in row 3, columns 10 to 12 to support path monitoring and one additional bit of path monitoring is defined in row 2, column 3, bit 7.

The ODU_k contains one instance of ODU PM overhead. The ODU_{Cn} contains n instances of the ODU PM overhead, numbered 1 to n (PM #1 to PM #n).

The PM and PM #1 field contains the following subfields (see Figure 15-18):

- trail trace identifier (TTI);
- bit interleaved parity (BIP-8);
- backward defect indication (BDI);
- backward error indication (BEI);
- status bits indicating the presence of a maintenance signal (STAT).

The PM #2 to #n fields contains the following subfields (see Figure 15-18):

- bit interleaved parity (BIP-8)
- backward error indication (BEI)
- reserved (RES).

The PM&TCM field contains the following PM subfield (see Figure 15-18):

- path delay measurement (DMp).

For the case of ODU_k, the content of the PM field, except the STAT subfield, will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODU-AIS, ODU-OCI, ODU-LCK). The content of the PM&TCM field will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal. Refer to clause 16.5.

For the case of ODU_{Cn}, the content of the PM field, except the STAT subfield, will be undefined (pattern will be all-1s or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODU_{Cn}-AIS, ODU_{Cn}-LCK). Refer to clause 16.5.

15.8.2.1.1 ODU PM trail trace identifier (TTI)

For path monitoring, a one-byte trail trace identifier (TTI) overhead is defined to transport the 64-byte TTI signal specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1].

The ODU_k and ODU_{Cn} contain one instance of ODU PM TTI overhead.

The 64-byte TTI signal shall be aligned with the ODU multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at ODU multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.8.2.1.2 ODU PM error detection code (BIP-8)

For path monitoring, a one-byte error detection code signal is defined in the ODU PM overhead. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation BIP-8 refers only to the number of BIP bits and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8, refer to the BIP-X definition in [ITU-T G.707].

Each ODU BIP-8 is computed over the bits in the OPU (columns 15 to 3824) area of ODU frame i, and inserted in the ODU PM BIP-8 overhead location in the ODU frame i+2 (see Figure 15-20).

The ODU_k contains one instance of ODU PM BIP-8 overhead. The ODU_{Cn} contains n instances of the ODU PM BIP-8 overhead, numbered 1 to n (BIP-8 #1 to BIP-8 #n).

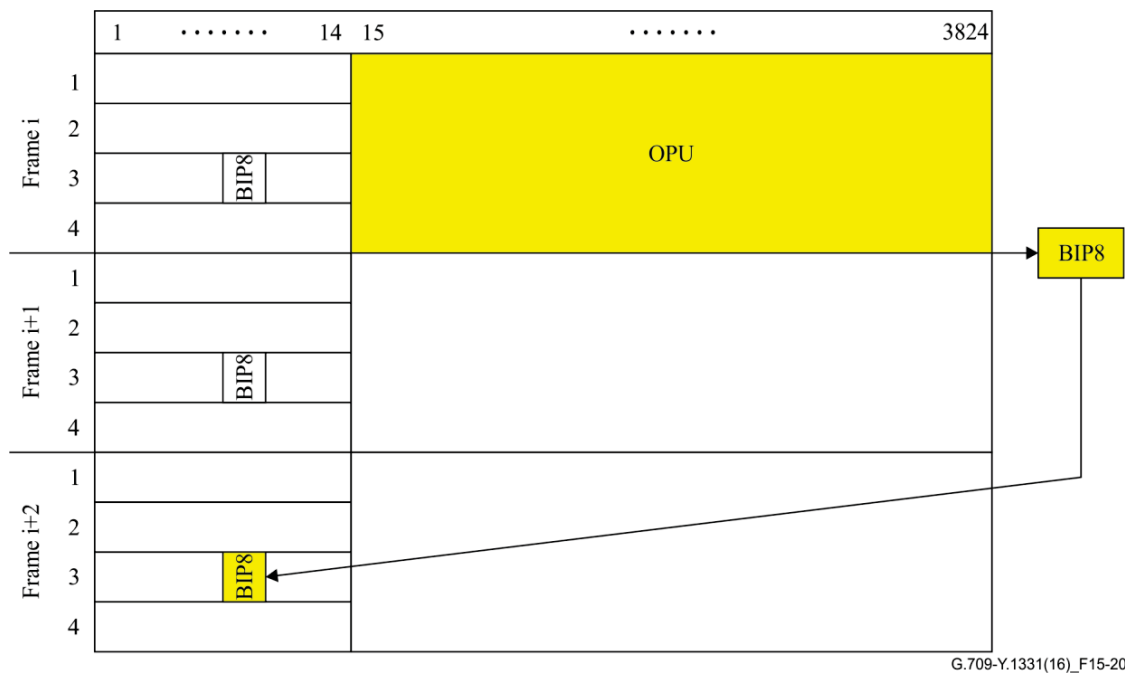


Figure 15-20 – ODU PM BIP-8 computation

15.8.2.1.3 ODU PM backward defect indication (BDI)

For path monitoring, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a path termination sink function in the upstream direction.

BDI is set to "1" to indicate an ODU backward defect indication, otherwise it is set to "0".

The ODU_k and ODU_{Cn} contain one instance of ODU PM BDI overhead.

15.8.2.1.4 ODU PM backward error indication (BEI)

For path monitoring, a four-bit backward error indication (BEI) signal is defined to convey in the upstream direction the count of interleaved-bit blocks that have been detected in error by the corresponding ODU path monitoring sink using the BIP-8 code. This count has nine legal values, namely 0-8 errors. The remaining seven possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-2).

The ODU_k contains one instance of ODU PM BEI overhead. The ODU_{Cn} contains n instances of the ODU PM BEI overhead, numbered 1 to n (BEI #1 to BEI #n).

Table 15-4 – ODU PM BEI interpretation

bits	ODU PM BEI	BIP violations
	1 2 3 4	
	0 0 0 0	0
	0 0 0 1	1
	0 0 1 0	2
	0 0 1 1	3
	0 1 0 0	4
	0 1 0 1	5
	0 1 1 0	6

Table 15-4 – ODU PM BEI interpretation

bits	ODU PM BEI 1 2 3 4	BIP violations
	0 1 1 1	7
	1 0 0 0	8
	1 0 0 1 to 1 1 1 1	0

15.8.2.1.5 ODU PM status (STAT)

For path monitoring, three bits are defined as status bits (STAT). They indicate the presence of a maintenance signal (see Table 15-3).

A P-CMEP sets these bits to "001".

The ODU_k and ODU_{Cn} contain one instance of ODU PM STAT overhead.

Table 15-5 – ODU PM status interpretation

bits	PM byte 3 6 7 8	Status
	0 0 0	Reserved for future international standardization
	0 0 1	Normal path signal
	0 1 0	Reserved for future international standardization
	0 1 1	Reserved for future international standardization
	1 0 0	Reserved for future international standardization
	1 0 1	Maintenance signal: ODU-LCK
	1 1 0	ODU _k : Maintenance signal: ODU-OCI ODU _{Cn} : Reserved for future international standardization
	1 1 1	Maintenance signal: ODU-AIS

15.8.2.1.6 ODU PM delay measurement (DMp)

For ODU path monitoring, a one-bit path delay measurement (DMp) signal is defined to convey the start of the delay measurement test.

The ODU_k and ODU_{Cn} contain one instance of ODU PM DMp overhead.

The DMp signal consists of a constant value (0 or 1) that is inverted at the beginning of a two-way delay measurement test. The transition from 0→1 in the sequence ...0000011111..., or the transition from 1→0 in the sequence ...1111100000... represents the path delay measurement start point. The new value of the DMp signal is maintained until the start of the next delay measurement test.

This DMp signal is inserted by the DMp originating P-CMEP and sent to the far-end P-CMEP. This far-end P-CMEP loops back the DMp signal towards the originating P-CMEP. The originating P-CMEP measures the number of frame periods between the moment the DMp signal value is inverted and the moment this inverted DMp signal value is received back from the far-end P-CMEP. The receiver should apply a persistency check on the received DMp signal to be tolerant for bit errors emulating the start of delay measurement indication. The additional frames that are used for such

persistence checking should not be added to the delay frame count. The looping P-CMEP should loop back each received DMp bit within approximately 100 μ s.

Refer to [ITU-T G.798] for the specific path delay measurement process specifications.

NOTE 1 – Path delay measurements can be performed on-demand, to provide the momentary two-way transfer delay status, and pro-active, to provide 15-minute and 24-hour two-way transfer delay performance management snapshots.

NOTE 2 – Equipment designed according to the 2008 or earlier versions of this Recommendation may not be capable of supporting this path delay monitoring. For such equipment, the DMp bit is a bit reserved for future international standardization and set to zero.

NOTE 3 – This process measures a round trip delay. The one way delay may not be half of the round trip delay in the case where the transmit and receive directions of the ODU network connection are of unequal lengths (e.g., in networks deploying unidirectional protection switching).

15.8.2.1.7 ODU PM reserved overhead (RES)

For path monitoring of the OTUCn, 12 bits in the PM overhead in the ODU OH #2 to #n are reserved for future international standardization. The value of these bits is set to "0".

The ODUk contains no ODU PM RES overhead. The ODUCn contains n-1 instances of the ODU PM RES overhead.

15.8.2.2 ODU tandem connection monitoring (TCM) overhead

Six fields of an ODU tandem connection monitoring (TCM) overhead are defined in row 2, columns 5 to 13 and row 3, columns 1 to 9 of the ODU overhead; and six additional bits of tandem connection monitoring are defined in row 2, column 3, bits 1 to 6.

TCM supports monitoring of ODUk connections for one or more of the following network applications (refer to [ITU-T G.805], [ITU-T G.872], [ITU-T G.873.2] and [ITU-T G.7714.1]):

- optical UNI-to-UNI tandem connection monitoring; monitoring the ODU connection through the public transport network (from public network ingress network termination to egress network termination);
- optical NNI-to-NNI tandem connection monitoring; monitoring the ODU connection through the network of a network operator (from operator network ingress network termination to egress network termination);
- sublayer monitoring for linear 1+1, 1:1 and 1:n ODUk subnetwork connection protection switching, to determine the signal fail and signal degrade conditions;
- sublayer monitoring for ODUk shared ring protection (SRP-1) protection switching as specified in [ITU-T G.873.2], to determine the signal fail and signal degrade conditions;
- sublayer monitoring for ODUk connection passing through two or more concatenated ODUk link connections (supported by back-to-back OTU trails), to provide a discovery message channel as specified in [ITU-T G.7714.1];
- monitoring an ODUk tandem connection for the purpose of detecting a signal fail or signal degrade condition in a switched ODUk connection, to initiate automatic restoration of the connection during fault and error conditions in the network;
- monitoring an ODUk tandem connection for, e.g., fault localization or verification of delivered quality of service.

TCM supports monitoring of segments of ODUCn connections which span multiple OTSiA subnetworks. The ODUCn TC-CMEP are located on OTUCn SOTU and MOTU interface ports at the edge of OTSiA subnetworks and/or on OTUCn SOTUm and MOTUm interface ports in ODUCn terminating nodes (e.g., in ODUk cross connects).

The six TCM fields are numbered TCM1, TCM2, ..., TCM6.

The ODU_k contains one instance of ODU TCM1 to TCM6 overhead. The ODU_{Cn} contains *n* instances of the ODU TCM1 to TCM6 overhead, numbered 1 to *n* (TCM_i #1 to TCM_i #*n*).

Each TCM_i and TCM_i #1 field contains the following subfields (see Figure 15-19):

- trail trace identifier (TTI);
- bit interleaved parity 8 (BIP-8);
- backward defect indication (BDI);
- backward error indication and backward incoming alignment error (BEI/BIAE);
- status bits indicating the presence of a TCM overhead, incoming alignment error, or a maintenance signal (STAT).

Each TCM_i #2 to #*n* field contains the following subfields (see Figure 15-19):

- bit interleaved parity 8 (BIP-8);
- backward error indication and backward incoming alignment error (BEI/BIAE);
- reserved (RES).

The PMandTCM field contains the following TCM subfields (see Figure 15-19):

- tandem connection delay measurement (DM_{ti}, *i*=1 to 6).

For the case of ODU_k, the content of the TCM fields, except the STAT subfield, will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODU_k-AIS, ODU_k-OCI, ODU_k-LCK). The content of the PM&TCM field will be undefined (pattern will be all-1s, 0110 0110 or 0101 0101 repeating) during the presence of a maintenance signal. Refer to clause 16.5.

For the case of ODU_{Cn}, the content of the TCM field groups, except the STAT subfield, will be undefined (pattern will be all-1s or 0101 0101 repeating) during the presence of a maintenance signal (e.g., ODU_{Cn}-AIS, ODU_{Cn}-LCK). The content of the PMand TCM field will be undefined (pattern will be all-1s or 0101 0101 repeating) during the presence of a maintenance signal. Refer to clause 6.5.

A TCM field and PM&TCM bit is assigned to a monitored connection as described in clause 15.8.2.2.6. The number of monitored connections along an ODU trail may vary between 0 and 6. These monitored connections may be nested, cascaded or both. Nesting and cascading are the default operational configurations. Overlapping is an additional configuration for testing purposes only. Overlapped monitored connections must be operated in a non-intrusive mode in which the maintenance signals ODU-AIS and ODU-LCK are not generated. For the case where one of the endpoints in an overlapping monitored connection is located inside an SNC protected domain while the other endpoint is located outside the protected domain, the SNC protection should be forced to working when the endpoint of the overlapping monitored connection is located on the working connection, and forced to protection when the endpoint is located on the protection connection.

Nesting and cascading configurations are shown in Figure 15-21. Monitored connections A1-A2/B1-B2/C1-C2 and A1-A2/B3-B4 are nested, while B1-B2/B3-B4 are cascaded. Overlapping is shown in Figure 15-22 (B1-B2 and C1-C2).

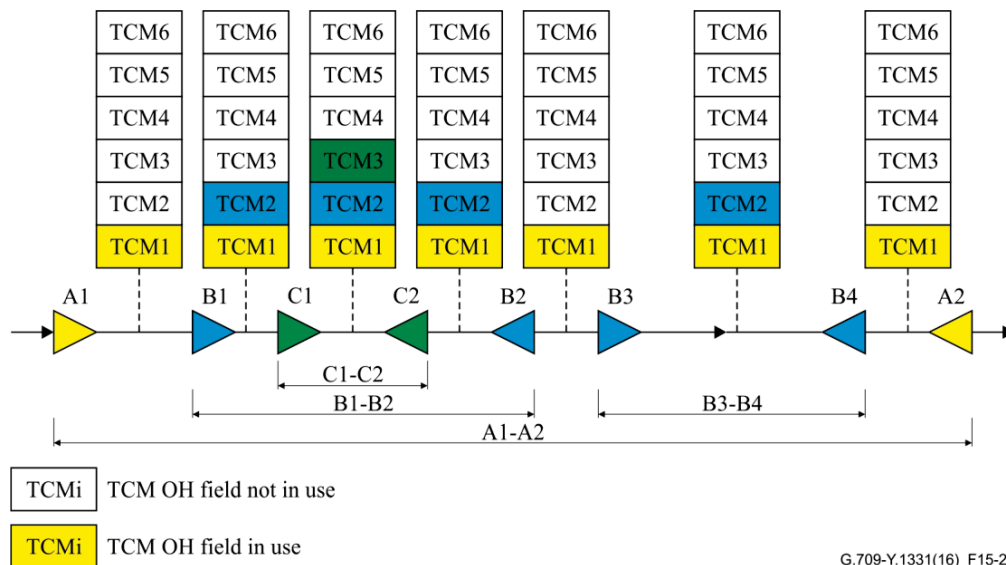


Figure 15-21 – Example of nested and cascaded ODU monitored connections

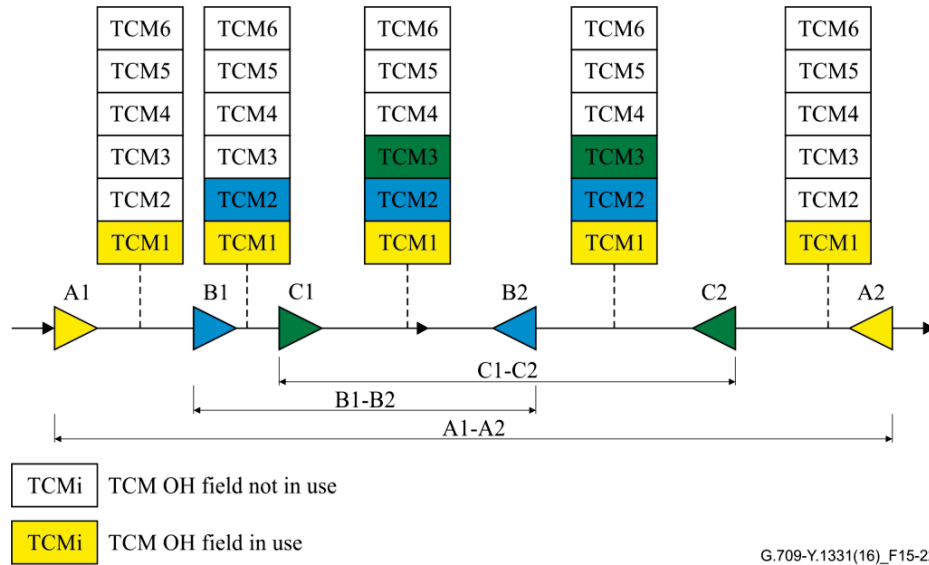


Figure 15-22 – Example of overlapping ODU monitored connections

15.8.2.2.1 ODU TCM trail trace identifier (TTI)

For each tandem connection monitoring field, one byte of overhead is allocated for the transport of the 64-byte trail trace identifier (TTI) specified in clause 15.2 or a discovery message as specified in [ITU-T G.7714.1] for TCM6.

The ODU_k and ODU_{Cn} contain one instance of ODU TTI overhead.

The 64-byte TTI signal shall be aligned with the ODU multiframe (see clause 15.6.2.2) and transmitted four times per multiframe. Byte 0 of the 64-byte TTI signal shall be present at ODU multiframe positions 0000 0000 (0x00), 0100 0000 (0x40), 1000 0000 (0x80) and 1100 0000 (0xC0).

15.8.2.2.2 ODU TCM error detection code (BIP-8)

For each tandem connection monitoring field, a one-byte error detection code signal is defined in the ODU TCM_i overhead. This byte provides a bit interleaved parity-8 (BIP-8) code.

NOTE – The notation *BIP-8* refers only to the number of BIP bits, and not to the EDC usage (i.e., what quantities are counted). For definition of BIP-8 refer to the BIP-X definition in [ITU-T G.707].

Each ODU TCM BIP-8 is computed over the bits in the OPU (columns 15 to 3824) area of ODU frame *i*, and inserted in the ODU TCM BIP-8 overhead location (associated with the tandem connection monitoring level) in ODU frame *i*+2 (see Figure 15-23).

The ODU_k contains one instance of ODU TCM_i BIP-8 overhead. The ODU_{Cn} contains *n* instances of the ODU TCM_i BIP-8 overhead, numbered 1 to *n* (BIP-8 #1 to BIP-8 #*n*).

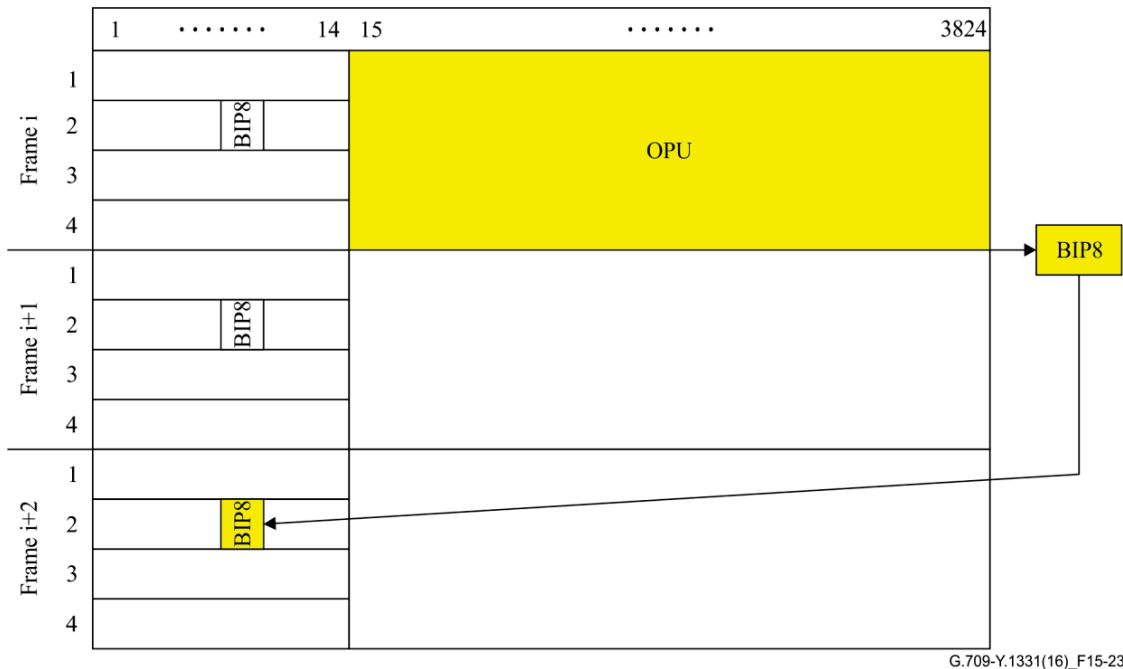


Figure 15-23 – ODU TCM BIP-8 computation

15.8.2.2.3 ODU TCM backward defect indication (BDI)

For each tandem connection monitoring field, a single-bit backward defect indication (BDI) signal is defined to convey the signal fail status detected in a tandem connection termination sink function in the upstream direction.

BDI is set to "1" to indicate an ODU_k backward defect indication; otherwise, it is set to "0".

The ODU_k and ODU_{Cn} contain one instance of ODU TCM_i BDI overhead.

15.8.2.2.4 ODU TCM backward error indication (BEI) and backward incoming alignment error (BIAE)

For each tandem connection monitoring field, a 4-bit backward error indication (BEI) and backward incoming alignment error (BIAE) signal is defined. This signal is used to convey in the upstream direction the count of interleaved-bit blocks that have been detected as being in error by the corresponding ODU tandem connection monitoring sink using the BIP-8 code. It is also used to convey in the upstream direction an incoming alignment error (IAE) condition that is detected in the corresponding ODU tandem connection monitoring sink in the IAE overhead.

During an IAE condition the code "1011" is inserted into the BEI/BIAE field and the error count is ignored. Otherwise the error count (0-8) is inserted into the BEI/BIAE field. The remaining six possible values represented by these four bits can only result from some unrelated condition and shall be interpreted as zero errors (see Table 15-4) and BIAE not active.

The ODU_k contains one instance of ODU TCM BEI/BIAE overhead. The ODU_{Cn} contains n instances of the ODU TCM_i BEI/BIAE overhead, numbered 1 to n (BEI/BIAE #1 to BEI/BIAE #n).

Table 15-6 – ODU TCM BEI/BIAE interpretation

ODU_k TCM BEI/BIAE bits	1 2 3 4	BIAE	BIP violations
0 0 0 0		false	0
0 0 0 1		false	1
0 0 1 0		false	2
0 0 1 1		false	3
0 1 0 0		false	4
0 1 0 1		false	5
0 1 1 0		false	6
0 1 1 1		false	7
1 0 0 0		false	8
1 0 0 1, 1 0 1 0		false	0
1 0 1 1		true	0
1 1 0 0 to 1 1 1 1		false	0

15.8.2.2.5 ODU TCM status (STAT)

For each tandem connection monitoring field, three bits are defined as status bits (STAT). They indicate the presence of a maintenance signal, if there is an incoming alignment error at the source TC-CMEP, or if there is no source TC-CMEP active (see Table 15-7).

The ODU_k and ODU_{Cn} contain one instance of ODU TCM STAT overhead.

Table 15-7 – ODU TCM status interpretation

TCM byte 3 bits	6 7 8	Status
0 0 0		No source TC
0 0 1		In use without IAE
0 1 0		In use with IAE
0 1 1		Reserved for future international standardization
1 0 0		Reserved for future international standardization
1 0 1		Maintenance signal: ODU-LCK
1 1 0		ODU _k : Maintenance signal: ODU _k -OCI ODU _{Cn} : Reserved for future international standardization
1 1 1		Maintenance signal: ODU-AIS

A P-CMEP sets these bits to "000".

A TC-CMEP ingress point sets these bits to either "001" to indicate to its peer TC-CMEP egress point that there is no incoming alignment error (IAE), or to "010" to indicate that there is an incoming alignment error.

The TC-CMEP egress point may use this information to suppress the counting of bit errors, which may occur as a result of a frame phase change of the ODU at the ingress of the tandem connection.

15.8.2.2.6 TCM overhead field assignment

Each TC-CMEP will be inserting/extracting its TCM overhead from one of the 6 TCM_i overhead fields and one of the 6 DMt_i fields. The specific TCM_i/DMt_i overhead field is provisioned by the network operator, network management system or switching control plane.

At a domain interface, it is possible to provision the maximum number (0 to 6) of tandem connection levels which will be passed through the domain. The default is three. These tandem connections should use the lower TCM_i/DMt_i overhead fields TCM₁/DMt₁...TCM_{MAX}/DMt_{MAX}. Overhead in TCM/DMt fields beyond the maximum (TCM_{max+1}/DMt_{max+1} and above) may/will be overwritten in the domain.

The TCM6 overhead field is assigned to monitor an ODUk connection which is supported by two or more concatenated ODUk link connections (supported by back-to-back OTUk trails). [ITU-T G.7714.1] specifies a discovery application which uses the TCM6 TTI SAPI field as discovery message channel. [ITU-T G.873.2] specifies an ODUk SRP-1 protection application which uses the TCM6 field to monitor the status/performance of the ODUk connection between two adjacent ODUk SRP-1 nodes.

Example

For the case of an ODUk leased circuit, the user may have been assigned one level of TCM, the service provider one level of TCM and each network operator (having a contract with the service provider) four levels of TCM. For the case where a network operator subcontracts part of its ODUk connection to another network operator, these four levels are to be split; e.g., two levels for the subcontracting operator.

This would result in the following TCM OH allocation:

- User: TCM1/DMt1 overhead field between the two user subnetworks, and TCM1/DMt1..TCM6/DMt6 within its own subnetwork;
- Service provider (SP): TCM2/DMt2 overhead field between two UNIs;
- Network operators NO1, NO2, NO3 having contract with service provider: TCM3/DMt3, TCM4/DMt4, TCM5/DMt5, TCM6/DMt6. Note that NO2 (which is subcontracting) cannot use TCM5/DMt5 and TCM6/DMt6 in the connection through the domain of NO4;
- NO4 (having subcontract with NO2): TCM5/DMt5, TCM6/DMt6.

See Figure 15-24.

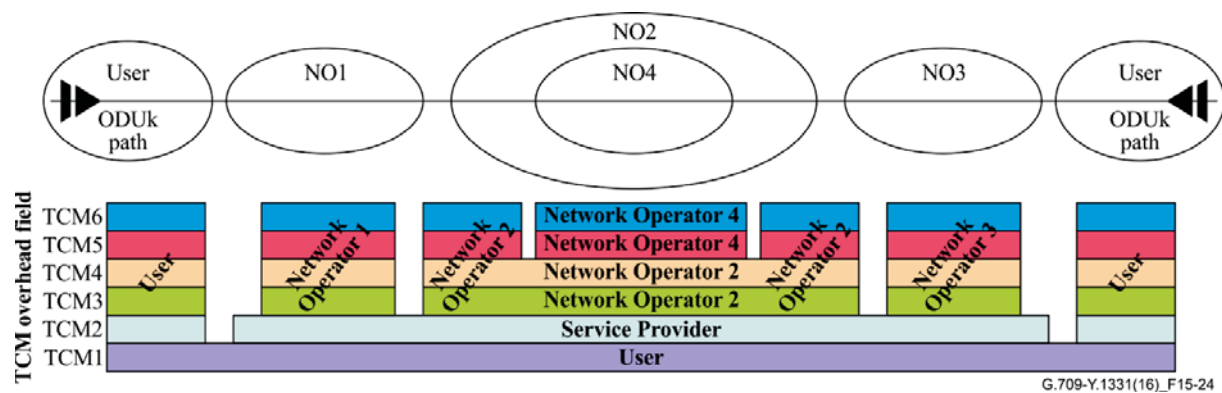


Figure 15-24 – Example of TCM overhead field assignment

15.8.2.2.7 Blank clause

This clause is intentionally left blank.

15.8.2.2.8 ODU TCM delay measurement (DMti, i=1 to 6)

For ODU tandem connection monitoring, a one-bit tandem connection delay measurement (DMti) signal is defined to convey the start of the delay measurement test.

The ODUk and ODUCn contain one instance of ODU TCM DMti overhead.

The DMti signal consists of a constant value (0 or 1) that is inverted at the beginning of a two-way delay measurement test. The transition from 0→1 in the sequence ...000001111..., or the transition from 1→0 in the sequence ...111110000... represents the path delay measurement start point. The new value of the DMti signal is maintained until the start of the next delay measurement test.

This DMti signal is inserted by the DMti originating TC-CMEP and sent to the far-end TC-CMEP. This far-end TC-CMEP loops back the DMti signal towards the originating TC-CMEP. The originating TC-CMEP measures the number of frame periods between the moment the DMti signal value is inverted and the moment this inverted DMti signal value is received back from the far-end TC-CMEP. The receiver should apply a persistency check on the received DMti signal to be tolerant for bit errors emulating the start of delay measurement indication. The additional frames that are used for such persistency checking should not be added to the delay frame count. The looping TC-CMEP should loop back each received DMti bit within approximately 100 μs.

Refer to [ITU-T G.798] for the specific tandem connection delay measurement process specifications.

NOTE 1 – Tandem connection delay measurements can be performed on-demand, to provide the momentary two-way transfer delay status, and pro-active, to provide 15-minute and 24-hour two-way transfer delay performance management snapshots.

NOTE 2 – Equipment designed according to the 2008 or earlier versions of this Recommendation may not be capable of supporting this tandem connection delay monitoring. For such equipment, the DMti bit is a bit reserved for future international standardization.

NOTE 3 – This process measures a round trip delay. The one way delay may not be half of the round trip delay in the case where the transmit and receive directions of the ODUk tandem connection are of unequal lengths (e.g., in networks deploying unidirectional protection switching).

15.8.2.2.9 ODU TCM reserved overhead (RES)

For tandem connection monitoring, 12 bits in the TCMi overhead are reserved for future international standardization in TCMi OH #2 to #n. The value of these bits is set to "0".

The ODUk contains no ODU TCMi RES overhead. The ODUCn contains n-1 instances of the ODU TCMi RES overhead.

15.8.2.3 ODU general communication channels (GCC1, GCC2)

Two fields of two bytes are allocated in the ODU overhead to support two general communications channels or two discovery channels as specified in [ITU-T G.7714.1] between any two network elements with access to the ODU frame structure (i.e., at 3R regeneration points).

These general communication channels are clear channels and any format specification is outside of the scope of this Recommendation. The bytes for GCC1 are located in row 4, columns 1 and 2, and the bytes for GCC2 are located in row 4, columns 3 and 4 of the ODU overhead.

The ODU_k contains one instance of ODU GCC1, GCC2 overhead. The ODU_{Cn} contains n instances of the ODU GCC1, GCC2 overhead, numbered 1 to n (GCC1 #1 to GCC1 #n, GCC2 #1 to GCC2 #n).

The GCC1 #1 to #n overhead are combined to provide one communication channel as illustrated in Figure 15-25 with an approximated bandwidth of $n \times 13.768$ Mbit/s.

The GCC2 #1 to #n overhead are combined to provide another communication channel as illustrated in Figure 15-25 with an approximated bandwidth of $n \times 13.768$ Mbit/s.

The GCC1 #1 to #n plus GCC2 #1 to #n overhead may be combined to provide one communication channel as illustrated in Figure 15-25 with an approximated bandwidth of $n \times 27.525$ Mbit/s.

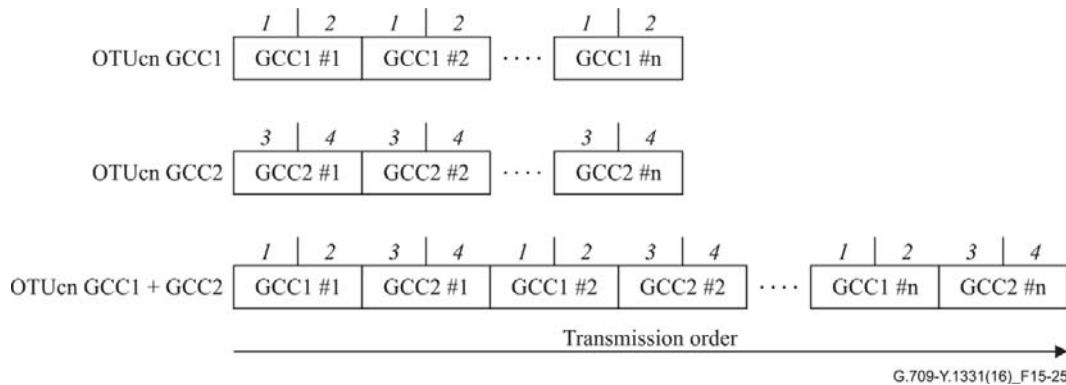


Figure 15-25 – OTUCn GCC1, GCC2 and GCC1+2 transmission order

15.8.2.4 ODU automatic protection switching and protection communication channel (APS/PCC)

A four-byte ODU-APS/PCC signal is defined in row 4, columns 5 to 8 of the ODU overhead. Up to eight levels of nested APS/PCC signals may be present in this field.

The ODU_k and ODU_{Cn} contain one instance of ODU APS/PCC overhead.

For ODU_k, the APS/PCC bytes in a given frame are assigned to a dedicated connection monitoring level depending on the value of MFAS as follows:

Table 15-8 – Multiframe to allow separate APS/PCC for each monitoring level

MFAS bits	6 7 8	APS/PCC channel applies to connection monitoring level	Protection scheme using the APS/PCC channel (Note 1)
0 0 0		ODU _k Path	ODU _k SNC/Ne, ODU _j CL-SNCG/I, Client SNC/I, ODU SRP-p
0 0 1		ODU _k TCM1	ODU _k SNC/S, ODU _k SNC/Ns
0 1 0		ODU _k TCM2	ODU _k SNC/S, ODU _k SNC/Ns
0 1 1		ODU _k TCM3	ODU _k SNC/S, ODU _k SNC/Ns

Table 15-8 – Multiframe to allow separate APS/PCC for each monitoring level

MFAS bits	6 7 8	APS/PCC channel applies to connection monitoring level	Protection scheme using the APS/PCC channel (Note 1)
1 0 0		ODUk TCM4	ODUk SNC/S, ODUk SNC/Ns
1 0 1		ODUk TCM5	ODUk SNC/S, ODUk SNC/Ns
1 1 0		ODUk TCM6	ODUk SNC/S, ODUk SNC/Ns, ODU SRP-1
1 1 1		ODUk server layer trail (Note 2)	ODUk SNC/I
<p>NOTE 1 – An APS channel may be used by more than one protection scheme and/or protection scheme instance. In case of nested protection schemes, care should be taken when an ODUk protection is to be set up in order not to interfere with the APS channel usage of another ODUk protection on the same connection monitoring level, e.g., protection can only be activated if that APS channel of the level is not already being used.</p> <p>NOTE 2 – Examples of ODUk server layer trails are an OTUk or a server ODUk (e.g., an ODU3 transporting an ODU1).</p>			

For ODU_{Cn}, the APS/PCC signal is used to support coordination of the end points in linear (ODUk CL-SNCG/I) and ring (ODUk SRP) protection applications.

For linear protection schemes, the bit assignments for these bytes and the bit-oriented protocol are given in [ITU-T G.873.1]. Bit assignment and byte-oriented protocol for ring protection schemes are given in [ITU-T G.873.2].

15.8.2.5 Blank clause

This clause is intentionally left blank.

15.8.2.6 ODU experimental overhead (EXP)

Four bytes are allocated in the ODU overhead for experimental use. These bytes are located in row 2, columns 4 and 14 and row 3, columns 13 and 14 of the ODU overhead.

The ODUk contains one instance of ODU EXP overhead. The ODU_{Cn} contains n instances of the ODU EXP overhead, numbered 1 to n (EXP #1 to EXP #n).

The use of these bytes is not subject to standardization and outside the scope of this Recommendation.

An experimental overhead is provided in the ODU OH to allow a vendor and/or a network operator within their own (sub)network to support an application, which requires an additional ODU overhead.

There is no requirement to forward the EXP overhead beyond the (sub)network; i.e., the operational span of the EXP overhead is limited to the (sub)network with the vendor's equipment, or the network of the operator.

15.8.2.7 ODU reserved overhead (RES)

For the case of an ODUk, eight bytes and one bit are reserved in the ODU overhead for future international standardization. These bytes are located in row 2, columns 1 to 2 and row 4, columns 9 to 14 of the ODU overhead. The bit is located in row 2, column 3, bit 8 of the ODU overhead. These bytes are set to all-0s.

For the case of an ODU_{Cn}, eight bytes and one bit in the ODU OH #1 and thirteen bytes in the ODU OH #2 to #n are reserved for future international standardization. These bytes are located in row 2, columns 1 to 2 and row 4, columns 9 to 14 of the ODU OH #1 and in row 2, columns 1 to 3 and row 4, columns 5 to 14 of the ODU OH #2 to #n. The bit is located in row 2, column 3, bit 8 of the ODU OH #1. These bytes and bit are set to all-0s.

15.9 OPU OH description

15.9.1 OPU OH location

The OPU overhead consists of: payload structure identifier (PSI) including the payload type (PT), the overhead associated with concatenation, the overhead associated with the mapping of client signals into the OPU payload and the overhead associated with the hitless adjustment of ODUflex client signals. The OPU PSI and PT overhead locations are shown in Figure 15-26.

The OPU_k contains one instance of OPU overhead. The OPU_{Cn} contains n instances of OPU overhead, numbered 1 to n (OPU OH #1 to #n).

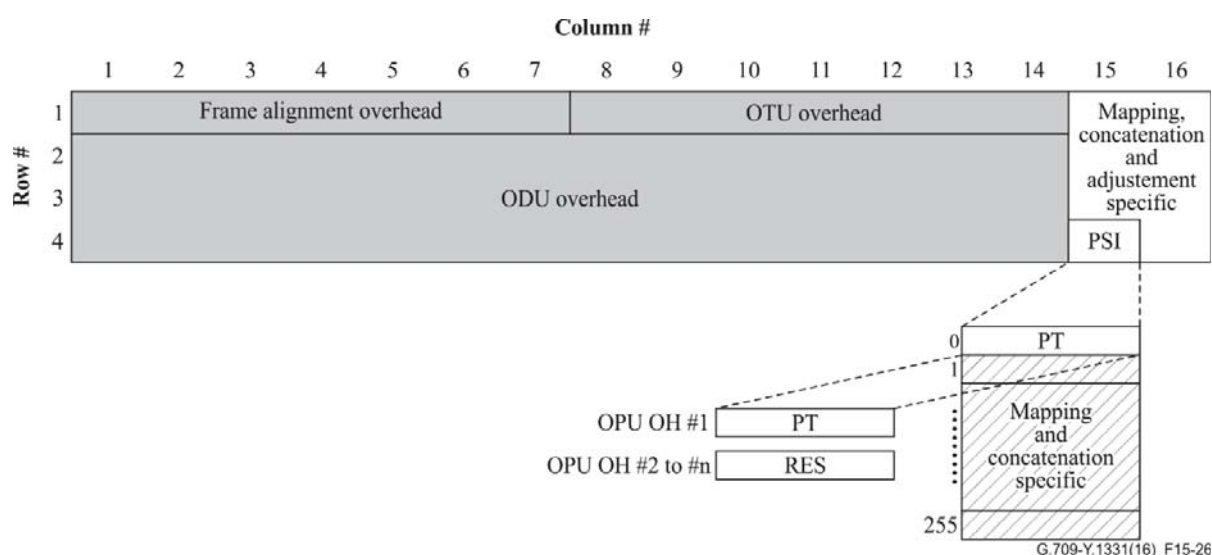


Figure 15-26 – OPU overhead

15.9.2 OPU OH definition

15.9.2.1 OPU payload structure identifier (PSI)

One byte is allocated in the OPU overhead to transport a 256-byte payload structure identifier (PSI) signal. The byte is located in row 4, column 15 of the OPU overhead.

The OPU_k contains one instance of OPU PSI overhead. The OPU_{Cn} contains n instances of the OPU PSI overhead, numbered 1 to n (PSI #1 to PSI #n).

The 256-byte PSI signal is aligned with the ODU multiframe (i.e., PSI[0] is present at ODU multiframe position 0000 0000, PSI[1] at position 0000 0001, PSI[2] at position 0000 0010, etc.).

PSI[0] contains a one-byte payload type or is reserved for future international standardization. PSI[1] to PSI[255] are mapping and concatenation specific, except for PT 0x01 (experimental mapping) and PTs 80-0x8F (for proprietary use).

15.9.2.1.1 OPU payload type (PT)

A one-byte payload type signal is defined in the PSI[0] byte of the payload structure identifier to indicate the composition of the OPU signal. The code points are defined in Table 15-9.

The OPU_k and OPU_{Cn} contain one instance of OPU PT overhead.

Table 15-9 – Payload type code points

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 0 0	0 0 0 1	01	Experimental mapping (Note 3)
0 0 0 0	0 0 1 0	02	Asynchronous CBR mapping, see clause 17.2
0 0 0 0	0 0 1 1	03	Bit-synchronous CBR mapping, see clause 17.2
0 0 0 0	0 1 0 0	04	Not available (Note 2)
0 0 0 0	0 1 0 1	05	GFP mapping, see clause 17.4
0 0 0 0	0 1 1 0	06	Not available (Note 2)
0 0 0 0	0 1 1 1	07	PCS codeword transparent Ethernet mapping: 1000BASE-X into OPU0, see clauses 17.7.1 and 17.7.1.1 40GBASE-R into OPU3, see clauses 17.7.4 and 17.7.4.1 100GBASE-R into OPU4, see clauses 17.7.5 and 17.7.5.1
0 0 0 0	1 0 0 0	08	FC-1200 into OPU2e mapping, see clause 17.8.2
0 0 0 0	1 0 0 1	09	GFP mapping into extended OPU2 payload, see clause 17.4.1 (Note 5)
0 0 0 0	1 0 1 0	0A	STM-1 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 0 1 1	0B	STM-4 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 1 0 0	0C	FC-100 mapping into OPU0, see clause 17.7.1
0 0 0 0	1 1 0 1	0D	FC-200 mapping into OPU1, see clause 17.7.2
0 0 0 0	1 1 1 0	0E	FC-400 mapping into OPUflex, see clause 17.9
0 0 0 0	1 1 1 1	0F	FC-800 mapping into OPUflex, see clause 17.9
0 0 0 1	0 0 0 0	10	Bit stream with octet timing mapping, see clause 17.6.1
0 0 0 1	0 0 0 1	11	Bit stream without octet timing mapping, see clause 17.6.2
0 0 0 1	0 0 1 0	12	IB SDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 0 1 1	13	IB DDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 1 0 0	14	IB QDR mapping into OPUflex, see clause 17.9
0 0 0 1	0 1 0 1	15	SDI mapping into OPU0, see clause 17.7.1
0 0 0 1	0 1 1 0	16	(1.485/1.001) Gbit/s SDI mapping into OPU1, see clause 17.7.2
0 0 0 1	0 1 1 1	17	1.485 Gbit/s SDI mapping into OPU1, see clause 17.7.2
0 0 0 1	1 0 0 0	18	(2.970/1.001) Gbit/s SDI mapping into OPUflex, see clause 17.9
0 0 0 1	1 0 0 1	19	2.970 Gbit/s SDI mapping into OPUflex, see clause 17.9
0 0 0 1	1 0 1 0	1A	SBCON/ESCON mapping into OPU0, see clause 17.7.1
0 0 0 1	1 0 1 1	1B	DVB_ASI mapping into OPU0, see clause 17.7.1
0 0 0 1	1 1 0 0	1C	FC-1600 mapping into OPUflex, see clause 17.9
0 0 0 1	1 1 0 1	1D	IMP mapping, see clause 17.11
0 0 0 1	1 1 1 0	1E	FlexE aware (partial rate) mapping into OPUflex, see clause 17.12
0 0 0 1	1 1 1 1	1F	FC-3200 mapping into OPUflex, see clause 17.9

Table 15-9 – Payload type code points

MSB 1 2 3 4	LSB 5 6 7 8	Hex code (Note 1)	Interpretation
0 0 1 0	0 0 0 0	20	ODU multiplex structure supporting ODTUjk only, see clause 19 (AMP only)
0 0 1 0	0 0 0 1	21	ODU multiplex structure supporting ODTUk.ts or ODTUk.ts and ODTUjk, see clause 19 (GMP capable) (Note 6)
0 0 1 0	0 0 1 0	22	ODU multiplex structure supporting ODTUCn.ts, see clause 20 (GMP capable)
0 1 0 1	0 1 0 1	55	Not available (Note 2)
0 1 1 0	0 1 1 0	66	Not available (Note 2)
1 0 0 0	x x x x	80-8F	Reserved codes for proprietary use (Note 4)
1 1 1 1	1 1 0 1	FD	NULL test signal mapping, see clause 17.5.1
1 1 1 1	1 1 1 0	FE	PRBS test signal mapping, see clause 17.5.2
1 1 1 1	1 1 1 1	FF	Not available (Note 2)
<p>NOTE 1 – There are 201 spare codes left for future international standardization. Refer to Annex A of [ITU-T G.806] for the procedure to obtain one of these codes for a new payload type.</p> <p>NOTE 2 – These values are excluded from the set of available code points. These bit patterns are present in ODUk maintenance signals or were used to represent client types that are no longer supported.</p> <p>NOTE 3 – Value "01" is only to be used for experimental activities in cases where a mapping code is not defined in this table. Refer to Annex A of [ITU-T G.806] for more information on the use of this code.</p> <p>NOTE 4 – These 16 code values will not be subject to further standardization. Refer to Annex A of [ITU-T G.806] for more information on the use of these codes.</p> <p>NOTE 5 – Supplement 43 (2008) to the ITU-T G-series of Recommendations indicated that this mapping recommended using payload type 87.</p> <p>NOTE 6 – Equipment supporting ODTUk.ts for OPU2 or OPU3 must be backward compatible with equipment which supports only the ODTUjk. ODTUk.ts capable equipment transmitting PT=21 which receives PT=20 from the far end shall revert to PT=20 and operate in ODTUjk only mode. Refer to [ITU-T G.798] for the specification.</p>			

15.9.2.2 OPU mapping specific overhead

Seven bytes are reserved in the OPUk overhead for the mapping, concatenation and hitless adjustment specific overhead. These bytes are located in rows 1 to 3, columns 15 and 16 and column 16 row 4. In addition, 255 bytes in the PSI #1 and 256 byte in PSI #2 to #n are reserved for mapping and concatenation specific purposes.

The use of these bytes depends on the specific client signal mapping (defined in clauses 17, 19 and 20), the use of concatenation (see clause 18) and use of hitless adjustment of ODUflex(GFP) (see [ITU-T G.7044]).

16 Maintenance signals

An alarm indication signal (AIS) is a signal sent downstream as an indication that an upstream defect has been detected. An AIS signal is generated in an adaptation sink function. An AIS signal is detected in a trail termination sink function to suppress defects or failures that would otherwise be detected as a consequence of the interruption of the transport of the original signal at an upstream point.

A forward defect indication (FDI) is a signal sent downstream as an indication that an upstream defect has been detected. An FDI signal is generated in an adaptation sink function. An FDI signal is detected

in a trail termination sink function to suppress defects or failures that would otherwise be detected as a consequence of the interruption of the transport of the original signal at an upstream point.

NOTE – AIS and FDI are similar signals. AIS is used as the term when the signal is in the digital domain. FDI is used as the term when the signal is in the optical domain; FDI is transported as a non-associated overhead.

An open connection indication (OCI) is a signal sent downstream as an indication that upstream the signal is not connected to a trail termination source. An OCI signal is generated in a connection function and output by this connection function on each of its output connection points, which are not connected to one of its input connection points. An OCI signal is detected in a trail termination sink function.

A locked (LCK) is a signal sent downstream as an indication that upstream the connection is "locked", and no signal has passed through.

A payload missing indication (PMI) is a signal sent downstream as an indication that upstream at the source point of the signal none of the frequency slots have an optical tributary signal. This indicates that the transport of the optical tributary signals is interrupted.

A PMI signal is generated in the adaptation source function and it is detected in the trail termination sink function which suppresses the LOS defect that arises under this condition.

16.1 OTS maintenance signals

16.1.1 OTS payload missing indication (OTS-PMI)

OTS-PMI is generated as an indication that the OTS payload does not contain an optical signal.

16.2 OMS maintenance signals

Three OMS maintenance signals are defined: OMS-FDI-P, OMS-FDI-O and OMS-PMI.

16.2.1 OMS forward defect indication – Payload (OMS-FDI-P)

OMS-FDI-P is generated as an indication of an OMS server layer defect in the OTS network layer.

16.2.2 OMS forward defect indication – Overhead (OMS-FDI-O)

OMS-FDI-O is generated as an indication when the transport of OMS OH via the OSC is interrupted due to a signal fail condition in the OSC.

16.2.3 OMS payload missing indication (OMS-PMI)

OMS-PMI is generated as an indication when none of the frequency slots contain an optical tributary signal.

16.3 OCh and OTiSA maintenance signals

Three OCh and OTiSA maintenance signals are defined: OCh-FDI-P, OCh-FDI-O and OCh-OCI.

16.3.1 OCh and OTiSA forward defect indication – Payload (OCh-FDI-P, OTSiA-FDI-P)

OCh-FDI-P and OCh-FDI-P are generated as an indication for an OCh and OTSiA server layer defect in the OMS network layer.

When the OTUk or OTUCn is terminated, the OCh-FDI-P and OTSiA_FDI-P is continued as an ODUk-AIS signal.

When the OTUCn is not terminated, the OCh-FDI-P and OTSiA_FDI-P are continued as an OTUCn-AIS signal.

16.3.2 OCh and OTiSA forward defect indication – Overhead (OCh-FDI-O, OTSiA-FDI-O)

OCh-FDI-O is generated as an indication when the transport of OCh OH via the OSC or OCC is interrupted due to a signal fail condition in the OSC or OCC.

16.3.3 OCh and OTiSA open connection indication (OCh-OCI, OTSiA-OCI)

The OCh-OCI/OTSiA-OCI signal indicates to downstream transport processing functions that the OCh/OTSiA connection is not bound to, or not connected (via a matrix connection) to a termination source function. The indication is used in order to distinguish downstream between a missing OCh/OTSiA signal due to a defect or due to the open connection (resulting from a management command).

NOTE – OCI is detected at the next downstream OCh or OTSiA trail terminating equipment. If the connection was opened intentionally, the related alarm report from this trail termination should be disabled by using the alarm reporting control mode (refer to [ITU-T M.3100]).

16.4 OTU maintenance signals

16.4.1 OTUk alarm indication signal (OTUk-AIS)

The OTUk-AIS (see Figure 16-1) is a generic-AIS signal (see clause 16.6.1). Since the OTUk capacity (130 560 bits) is not an integer multiple of the PN-11 sequence length (2047 bits), the PN-11 sequence may cross an OTUk frame boundary.

NOTE – OTUk-AIS is defined to support a future server layer application. OTN equipment should be capable of detecting the presence of such a signal within OTUk (k=1,2,3) SOTU and OTUk (k=1,2) MOTU interface signals; it is not required to generate such a signal.

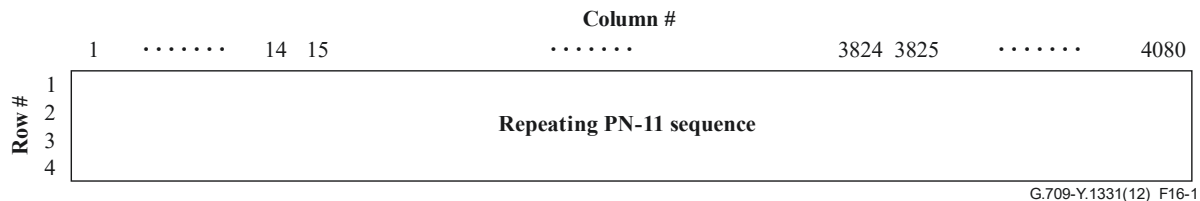


Figure 16-1 – OTUk-AIS

16.4.2 OTUCn alarm indication signal (OTUCn-AIS)

The OTUCn-AIS is specified as all "1"s in the entire OTUCn signal, excluding the frame alignment overhead (FA OH) (see Figure 16-2).

The presence of the OTUCn-AIS is detected by monitoring the OTUCn SM STAT bits in the SM overhead fields.

NOTE – OTUCn-AIS is defined to support a future 3R regenerator application in which the OTUCn is not terminated and an OTUCn subrating application. OTN equipment should be capable of detecting the presence of such a signal within OTUCn signals on SOTUm and MOTUm interfaces.

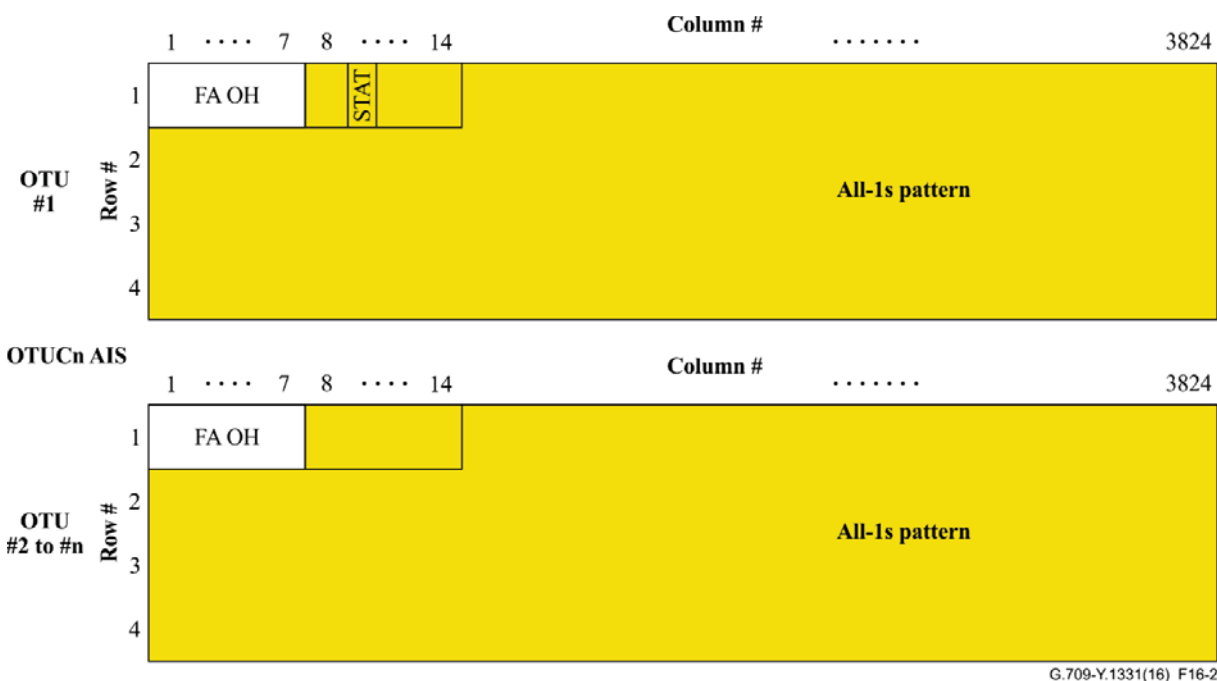


Figure 16-2 – OTUCn-AIS

16.5 ODU maintenance signals

Three ODU maintenance signals are defined: ODU-AIS, ODU-OCI and ODU-LCK.

16.5.1 ODU alarm indication signal (ODU-AIS)

ODUk-AIS is specified as all "1"s in the entire ODU signal, excluding the frame alignment overhead (FA OH), OTU overhead (OTU OH) (see Figure 16-3).

The ODUk contains one instance of ODU AIS. The ODUCn contains n instances of ODU AIS, numbered 1 to n (ODU AIS #1 to #n).

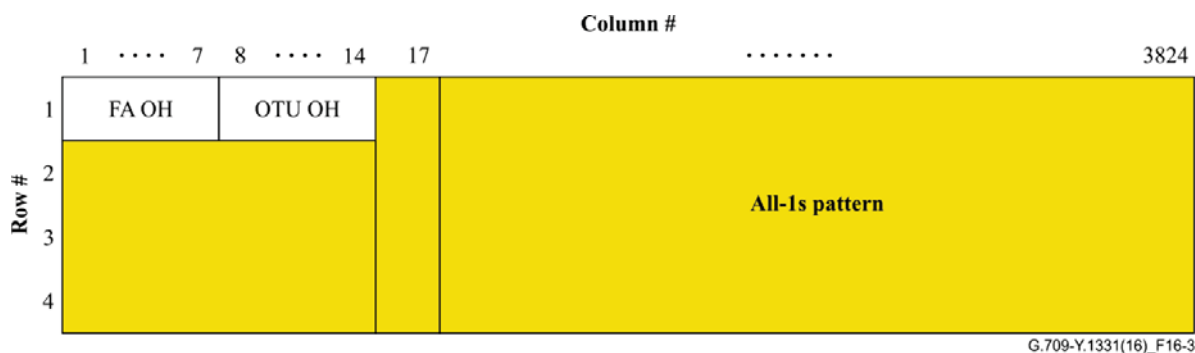


Figure 16-3 – ODU-AIS

In addition, the ODU-AIS signal may be extended with one or more levels of ODU tandem connection, GCC1, GCC2, EXP and/or APS/PCC overhead before it is presented at the OTN interface. This is dependent on the functionality between the ODU-AIS insertion point and the OTN interface.

The presence of the ODU-AIS is detected by monitoring the ODU STAT bits in the PM and TCMi overhead fields.

16.5.2 ODUk open connection indication (ODUk-OCI)

ODUk-OCI is specified as a repeating "0110 0110" pattern in the entire ODUk signal, excluding the frame alignment overhead (FA OH) and OTUk overhead (OTUk OH) (see Figure 16-4).

NOTE – ODUCn OCI is not defined.

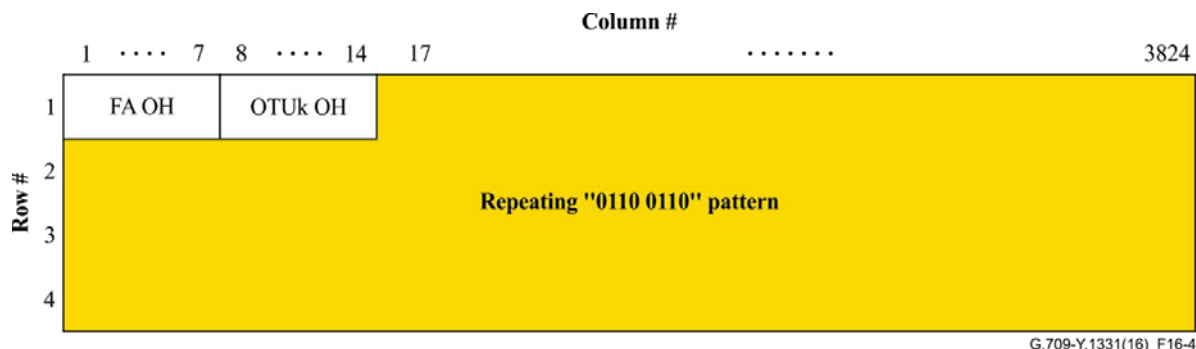


Figure 16-4 – ODUk-OCI

NOTE – The repeating "0110 0110" pattern is the default pattern; other patterns are also allowed as long as the STAT bits in the PM and TCMi overhead fields are set to "110".

In addition, the ODUk-OCI signal may be extended with one or more levels of ODUk tandem connection, GCC1, GCC2, EXP and/or APS/PCC overhead before it is presented at the OTM interface. This is dependent on the functionality between the ODUk-OCI insertion point and the OTM interface.

The presence of ODUk-OCI is detected by monitoring the ODUk STAT bits in the PM and TCMi overhead fields.

16.5.3 ODU locked (ODU-LCK)

ODU-LCK is specified as a repeating "0101 0101" pattern in the entire ODU signal, excluding the frame alignment overhead (FA OH) and OTU overhead (OTU OH) (see Figure 16-5).

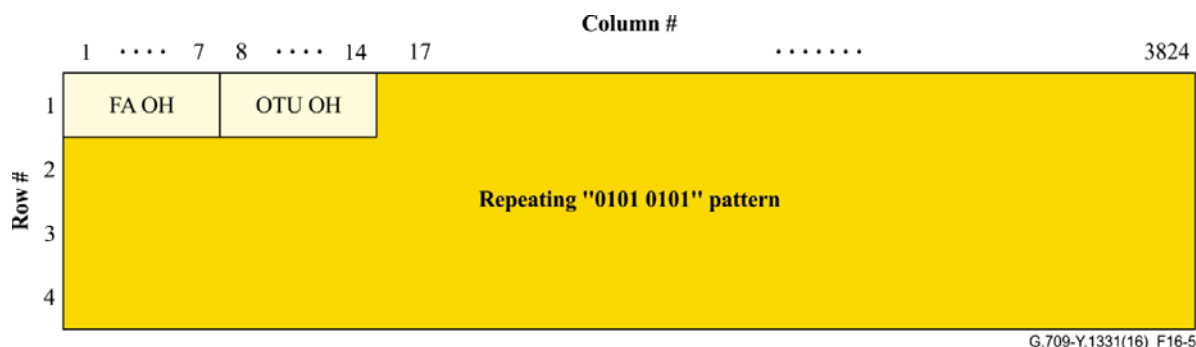


Figure 16-5 – ODU-LCK

NOTE – The repeating "0101 0101" pattern is the default pattern; other patterns are also allowed as long as the STAT bits in the PM and TCMi overhead fields are set to "101".

In addition, the ODU-LCK signal may be extended with one or more additional levels of ODU tandem connection, GCC1, GCC2, EXP and/or the APS/PCC overhead before it is presented at the OTN interface. This is dependent on the functionality between the ODU-LCK insertion point and the OTN interface.

The presence of ODU-LCK is detected by monitoring the ODU STAT bits in the PM and TCMi overhead fields.

16.6 Client maintenance signal

16.6.1 Generic AIS for constant bit rate signals

The generic-AIS signal is a signal with a 2 047-bit polynomial number 11 (PN-11) repeating sequence.

The PN-11 sequence is defined by the generating polynomial $1 + x^9 + x^{11}$ as specified in clause 5.2 of [ITU-T O.150]. (See Figure 16-6.)

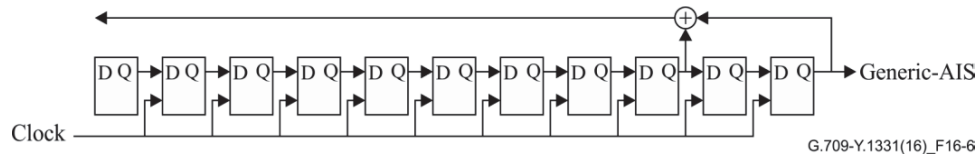


Figure 16-6 – Generic-AIS generating circuit

17 Mapping of client signals

This clause specifies the mapping of:

- STM-16, STM-64, STM-256 constant bit rate client signals into OPU_k using client/server specific asynchronous or bit-synchronous mapping procedures (AMP, BMP);
- 10GBASE-R constant bit rate client signal into OPU_{2e} using client/server specific bit-synchronous mapping procedure (BMP);
- FC-1200 constant bit rate client signal after timing transparent transcoding (TTT) providing a 50/51 rate compression into OPU_{2e} using client/server specific byte-synchronous mapping procedure;
- constant bit rate client signals with bit rates up to 1.238 Gbit/s into OPU₀ and up to 2.488 Gbit/s into OPU₁ using a client agnostic generic mapping procedure (GMP) possibly preceded by a timing transparent transcoding (TTT) of the client signal to reduce the bit rate of the signal to fit the OPU_k payload bandwidth;
- constant bit rate client signals into OPU₁, OPU₂, OPU₃ or OPU₄ respectively using a client agnostic generic mapping procedure (GMP) possibly preceded by a timing transparent transcoding (TTT) of the client signal to reduce the bit rate of the signal to fit the OPU_k payload bandwidth;
- other constant bit rate client signals into OPU_{flex} using a client agnostic bit-synchronous mapping procedure (BMP);
- packet streams (e.g., Ethernet, MPLS, IP) which are encapsulated with the generic framing procedure (GFP-F);
- test signals;
- continuous mode GPON constant bit rate client signal into OPU₁ using asynchronous mapping procedure (AMP);
- continuous mode XGPON constant bit rate client signal into OPU₂ using asynchronous mapping procedure (AMP);
- FlexE-aware client signal into OPU_{flex} using bitsynchronous mapping procedure (BMP);
- FlexE Client client signal into OPU_{flex} using idle mapping procedure;
- packet streams (e.g., Ethernet, MPLS, IP) which are encapsulated with the Idle mapping procedure (IMP)

into OPU.

17.1 OPU client signal fail (CSF)

For support of local management systems, a single-bit OPU client signal fail (CSF) indicator is defined to convey the signal fail status of the CBR and Ethernet private line client signal mapped into an OPU at the ingress of the OTN to the egress of the OTN.

OPU CSF is located in bit 1 of the PSI[2] byte of the payload structure identifier. Bits 2 to 8 of the PSI[2] byte are reserved for future international standardization. These bits are set to all-0s.

OPU CSF is set to "1" to indicate a client signal fail indication, otherwise it is set to "0".

NOTE – Equipment designed prior to Edition 3.0 of the Recommendation will generate a "0" in the OPUk CSF and will ignore any value in OPUk CSF.

17.2 Mapping of CBR2G5, CBR10G, CBR10G3 and CBR40G signals into OPUk

The mapping of a CBR2G5, CBR10G or CBR40G signal (with up to ± 20 ppm bit-rate tolerance) into an OPUk ($k = 1, 2, 3$) may be performed according to the bit-synchronous mapping procedure based on one generic OPUk frame structure (see Figure 17-1). The mapping of a CBR2G5, CBR10G or CBR40G signal (with up to ± 45 ppm bit-rate tolerance) into an OPUk ($k = 1, 2, 3$) may be performed according to the asynchronous mapping procedure. The mapping of a CBR10G3 signal (with up to ± 100 ppm bit-rate tolerance) into an OPUk ($k = 2e$) is performed using the bit-synchronous mapping procedure.

NOTE 1 – Examples of CBR2G5, CBR10G and CBR40G signals are STM-16 and CMGPON_D/U2 (refer to [ITU-T G.984.6]), STM-64 and CMXGPON_D/U2 [ITU-T G.987.4] and STM-256. An example of a CBR10G3 signal is 10GBASE-R.

NOTE 2 – The maximum bit-rate tolerance between an OPUk and the client signal clock, which can be accommodated by the asynchronous mapping scheme, is ± 65 ppm. With a bit-rate tolerance of ± 20 ppm for the OPUk clock, the client signal's bit-rate tolerance can be ± 45 ppm.

NOTE 3 – For OPUk ($k=1,2,3$) the clock tolerance is ± 20 ppm. For OPU2e the clock tolerance is ± 100 ppm and asynchronous mapping cannot be supported with this justification overhead.

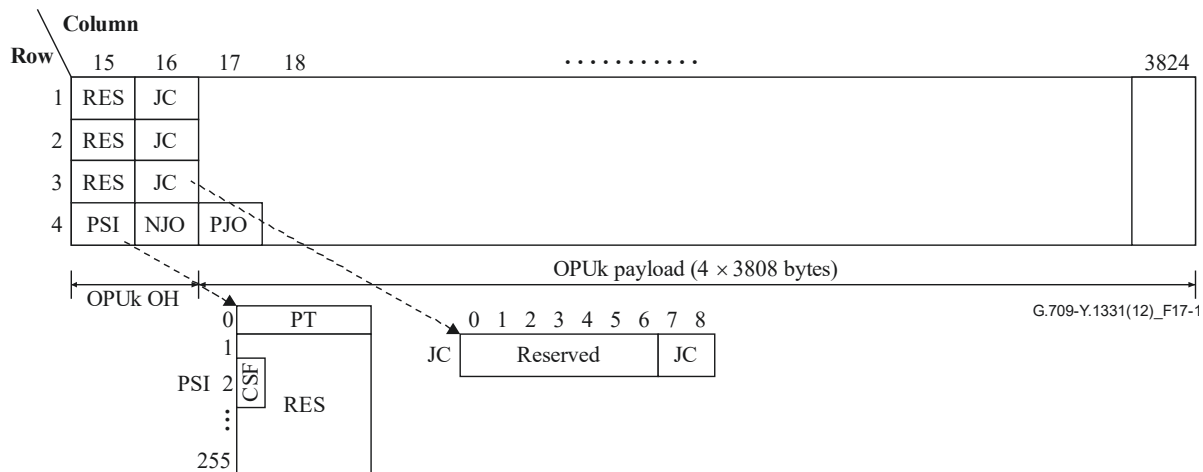


Figure 17-1 – OPUk frame structure for the mapping of a CBR2G5, CBR10G or CBR40G signal

The OPUk overhead for these mappings consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits reserved for future international standardization (RES), three justification control (JC) bytes, one negative justification opportunity (NJO) byte, and three bytes reserved for future international standardization (RES).

The JC bytes consist of two bits for justification control and six bits reserved for future international standardization.

The OPUk payload for these mappings consists of 4×3808 bytes, including one positive justification opportunity (PJO) byte.

The justification control (JC) signal, which is located in rows 1, 2 and 3 of column 16, bits 7 and 8, is used to control the two justification opportunity bytes NJO and PJO that follow in row 4.

The asynchronous and bit-synchronous mapping processes generate the JC, NJO and PJO according to Tables 17-1 and 17-2, respectively. The de-mapping process interprets JC, NJO and PJO according to Table 17-3. Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

Table 17-1 – JC, NJO and PJO generation by an asynchronous mapping process

bits	JC 7 8	NJO	PJO
	0 0	justification byte	data byte
	0 1	data byte	data byte
	1 0	not generated	
	1 1	justification byte	justification byte

Table 17-2 – JC, NJO and PJO generation by a bit-synchronous mapping process

bits	JC 7 8	NJO	PJO
	0 0	justification byte	data byte
	0 1	not generated	
	1 0		
	1 1		

Table 17-3 – JC, NJO and PJO interpretation

bits	JC 7 8	NJO	PJO
	0 0	justification byte	data byte
	0 1	data byte	data byte
	1 0 (Note)	justification byte	data byte
	1 1	justification byte	justification byte
NOTE – A mapper circuit does not generate this code. Due to bit errors a de-mapper circuit might receive this code.			

The value contained in NJO and PJO when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming CBR2G5, CBR10G or CBR40G client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the generic-AIS signal as specified in clause 16.6.1, and is then mapped into the OPUk.

During a signal fail condition of the incoming 10GBASE-R type CBR10G3 client signal (e.g., in the case of a loss of input signal), this failed incoming 10GBASE-R signal is replaced by a stream of 66B blocks, with each block carrying two local fault sequence ordered sets (as specified in [IEEE 802.3]). This replacement signal is then mapped into the OPU2e.

During the signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the generic-AIS pattern as specified in clause 16.6.1 is generated as a replacement signal for the lost CBR2G5, CBR10G or CBR40G signal.

During the signal fail condition of the incoming ODU2e/OPU2e signal (e.g., in the case of an ODU2e-AIS, ODU2e-LCK, ODU2e-OCI condition) a stream of 66B blocks, with each block carrying two local fault sequence ordered sets (as specified in [IEEE 802.3]) is generated as a replacement signal for the lost 10GBASE-R signal.

NOTE 4 – Local fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH=10 0x55 00 00 01 00 00 00 01.

NOTE 5 – Equipment developed prior to Edition 2.5 of this Recommendation may generate a different 10GBASE-R replacement signal (e.g., Generic-AIS) than the local fault sequence ordered set.

Asynchronous mapping

The OPUk signal for the asynchronous mapping is created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the CBR2G5, CBR10G or CBR40G (i.e., $4^{(k-1)} \times 2\,488\,320$ kbit/s ($k = 1,2,3$)) client signal.

The CBR2G5, CBR10G, CBR40G (i.e., $4^{(k-1)} \times 2\,488\,320$ kbit/s ($k = 1,2,3$)) signal is mapped into the OPUk using a positive/negative/zero (pnz) justification scheme.

Bit-synchronous mapping

The OPUk clock for bit-synchronous mapping is derived from the CBR2G5, CBR10G, CBR40G or CBR10G3 client signal. During signal fail conditions of the incoming CBR2G5, CBR10G, CBR40G or CBR10G3 signal (e.g., in the case of a loss of input signal), the OPUk payload signal bit rate shall be within the limits specified in Table 7-3 and neither an OPUk frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming CBR2G5, CBR10G, CBR40G or CBR10G3 signal shall be done without introducing an OPUk frequency or frame phase discontinuity.

The CBR2G5, CBR10G, CBR40G or CBR10G3 signal is mapped into the OPUk without using the justification capability within the OPUk frame: NJO contains a justification byte, PJO contains a data byte, and the JC signal is fixed to 00.

17.2.1 Mapping a CBR2G5 signal (e.g., STM-16, CMGPON_D/CMGPON_U2) into OPU1

Groups of eight successive bits (not necessarily being a byte) of the CBR2G5 signal are mapped into a data (D) byte of the OPU1 (see Figure 17-2). Once per OPU1 frame, it is possible to perform either a positive or a negative justification action.

Row #	15	16	17	18	Column #	3824

1	RES	JC	D	D	3805D	D
2	RES	JC	D	D	3805D	D
3	RES	JC	D	D	3805D	D
4	PSI	NJO	PJO	D	3805D	D

G.709-Y.1331(12)_F17-2

Figure 17-2 – Mapping of a CBR2G5 signal into OPU1

17.2.2 Mapping a CBR10G signal (e.g., STM-64, CMXGPON_D/CMXGPON_U2) into OPU2

Groups of eight successive bits (not necessarily being a byte) of the CBR10G signal are mapped into a data (D) byte of the OPU2 (see Figure 17-3). 64 fixed stuff (FS) bytes are added in columns 1905 to 1920. Once per OPU2 frame, it is possible to perform either a positive or a negative justification action.

		Column #										
		15	16	17	1904	1905	1920	1921	3824
Row #	1	RES	JC	118 × 16D					16FS		119 × 16D	
	2	RES	JC	118 × 16D					16FS		119 × 16D	
	3	RES	JC	118 × 16D					16FS		119 × 16D	
	4	PSI	NJO	PJO	15D + 117 × 16D					16FS		119 × 16D

G.709-Y.1331(12)_F17-3

Figure 17-3 – Mapping of a CBR10G signal into OPU2

17.2.3 Mapping a CBR40G signal (e.g., STM-256) into OPU3

Groups of eight successive bits (not necessarily being a byte) of the CBR40G signal are mapped into a data (D) byte of the OPU3 (see Figure 17-4). 128 fixed stuff (FS) bytes are added in columns 1265 to 1280 and 2545 to 2560. Once per OPU3 frame, it is possible to perform either a positive or a negative justification action.

		Column #																
		15	16	17	1264	1265	...	1280	1281	2544	2545	...	2560	2561	...	3824
Row #	1	RES	JC	78 × 16D			16FS			79 × 16D			16FS			79 × 16D		
	2	RES	JC	78 × 16D			16FS			79 × 16D			16FS			79 × 16D		
	3	RES	JC	78 × 16D			16FS			79 × 16D			16FS			79 × 16D		
	4	PSI	NJO	PJO	15D + 77 × 16D			16FS			79 × 16D			16FS			79 × 16D	

G.709-Y.1331(12)_F17-4

Figure 17-4 – Mapping of a CBR40G signal into OPU3

17.2.4 Mapping a CBR10G3 signal (e.g., 10GBASE-R) into OPU2e

Groups of eight successive bits (not necessarily being a byte) of the CBR10G3 signal are bit-synchronously mapped into a data (D) byte of the OPU2e (see Figure 17-5). 64 fixed stuff (FS) bytes are added in columns 1905 to 1920.

NOTE – The NJO byte will always carry a stuff byte, the PJO byte will always carry a data (D) byte and the JC bytes will always carry the all-0s pattern.

		Column #											
		15	16	17	1904	1905	1920	1921	3824	
Row #	1	RES	JC					118 × 16D		16FS		119 × 16D	
	2	RES	JC					118 × 16D		16FS		119 × 16D	
	3	RES	JC					118 × 16D		16FS		119 × 16D	
	4	PSI	NJO	PJO					15D + 117 × 16D		16FS		119 × 16D

G.709-Y.1331(12)_F17-5

Figure 17-5 – Mapping of a CBR10G3 signal into OPU2e

17.3 Blank clause

This clause is intentionally left blank.

17.4 Mapping of GFP frames into OPUk (k=0,1,2,3,4,flex)

The mapping of generic framing procedure (GFP) frames is performed by aligning the byte structure of every GFP frame with the byte structure of the OPUk payload (see Figure 17-6). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a frame may cross the OPUk (k=0,1,2,3,4,flex) frame boundary.

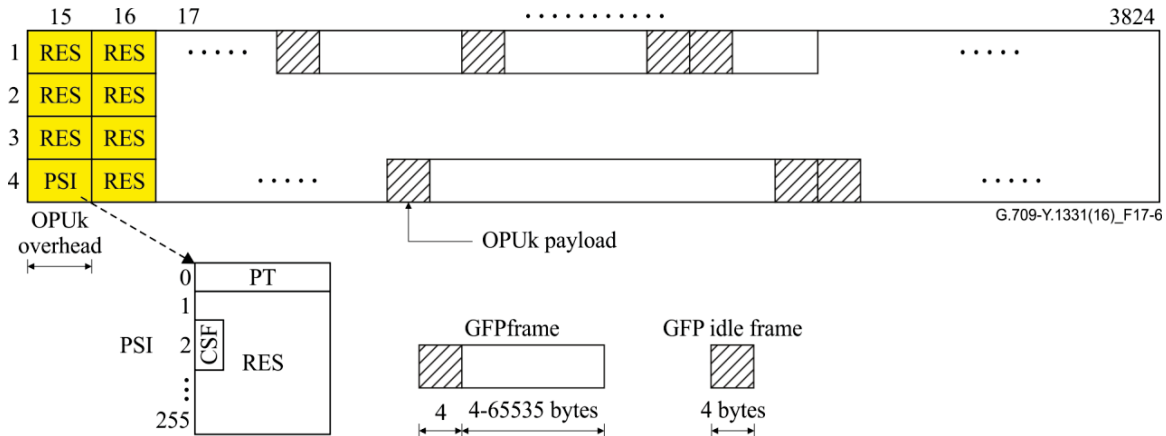


Figure 17-6 – OPUk frame structure and mapping of GFP frames into OPUk

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPUk payload area, due to the insertion of idle frames at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE 1 – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPUk overhead for the GFP mapping consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES). The CSF indicator should be used only for Ethernet private line type 1 services; for other packet clients the CSF bit is fixed to 0.

The OPUk payload for the GFP mapping consists of 4×3808 bytes.

NOTE 2 – The OPUflex(GFP) bit rate may be any configured bit rate as specified in Tables 7-3 and 7-8.

17.4.1 Mapping of GFP frames into an extended OPU2 payload area

The mapping of generic framing procedure (GFP) frames in an extended OPU2 payload area is performed by aligning the byte structure of every GFP frame with the byte structure of the extended OPU2 payload (see Figure 17-7). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a frame may cross the OPU2 frame boundary.

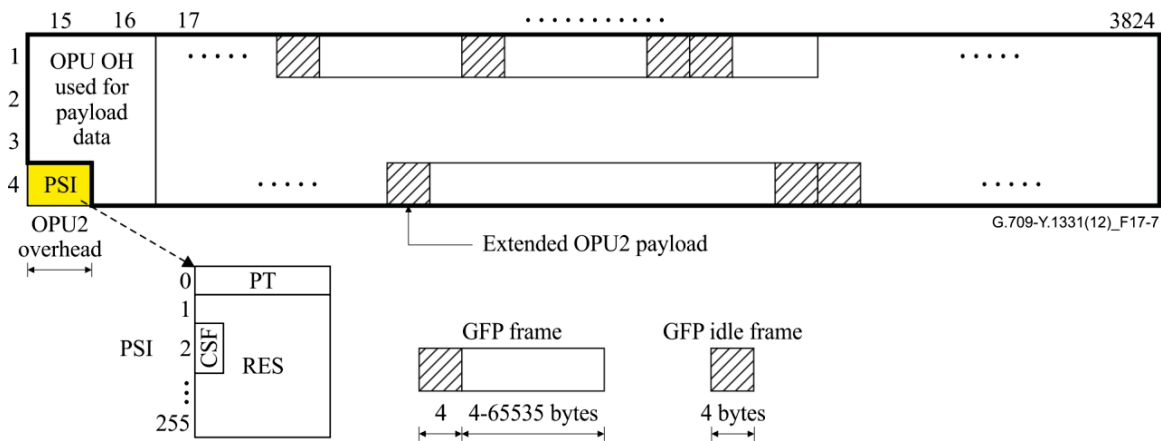


Figure 17-7 – OPU2 frame structure and mapping of GFP frames into an extended OPU2 payload area

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPU2 payload area, due to the insertion of GFP-idle frames at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPU2 overhead for the GFP mapping consists of a payload structure identifier (PSI) including the payload type (PT), a client signal fail (CSF) indicator and 254 bytes plus 7 bits of reserved for future international standardization (RES).

The extended OPU2 payload for the GFP mapping consists of 4×3808 bytes from the OPU2 payload plus 7 bytes from the OPU2 overhead.

17.5 Mapping of test signal into OPU

17.5.1 Mapping of a NULL client into OPU

An OPU payload signal with an all-0s pattern (see Figure 17-8) is defined for test purposes. This is referred to as the NULL client.

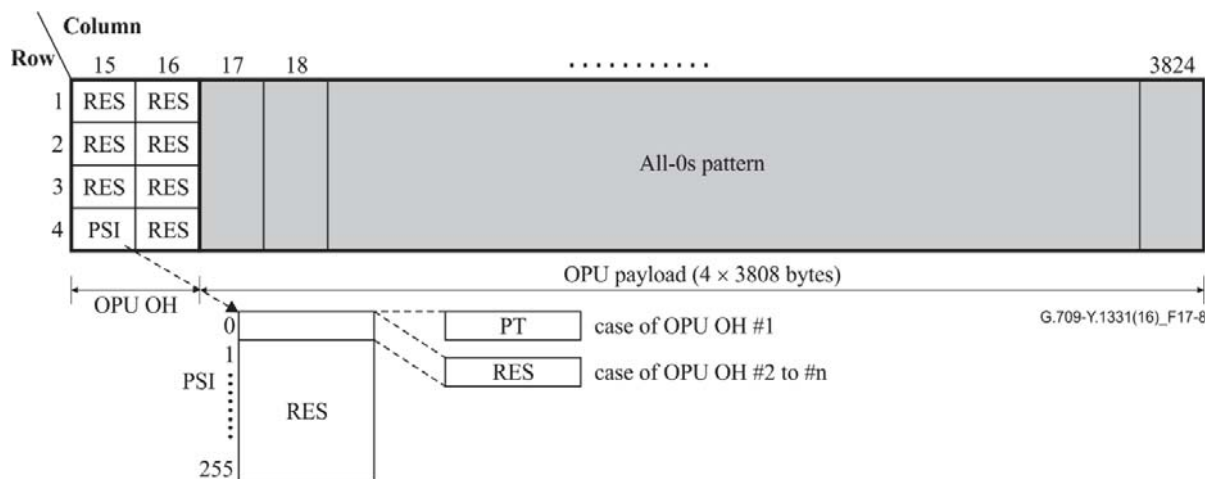


Figure 17-8 – OPU frame structure and mapping of a NULL client into OPU

The OPU overhead for the NULL mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES).

The OPU payload for the NULL mapping consists of 4×3808 bytes.

The OPU_k contains one instance of the NULL client. The OPU_{Cn} contains n instances of the NULL client, numbered 1 to n.

17.5.2 Mapping of PRBS test signal into OPU

For end-to-end and segment turn-up test purposes, a 2 147 483 647-bit pseudo-random test sequence ($2^{31} - 1$) as specified in clause 5.8 of [ITU-T O.150] can be mapped into the OPU payload. Groups of eight successive bits of the 2 147 483 647-bit pseudo-random test sequence signal are mapped into 8 data bits (8D) (i.e., one byte) of the OPU payload (see Figure 17-9).

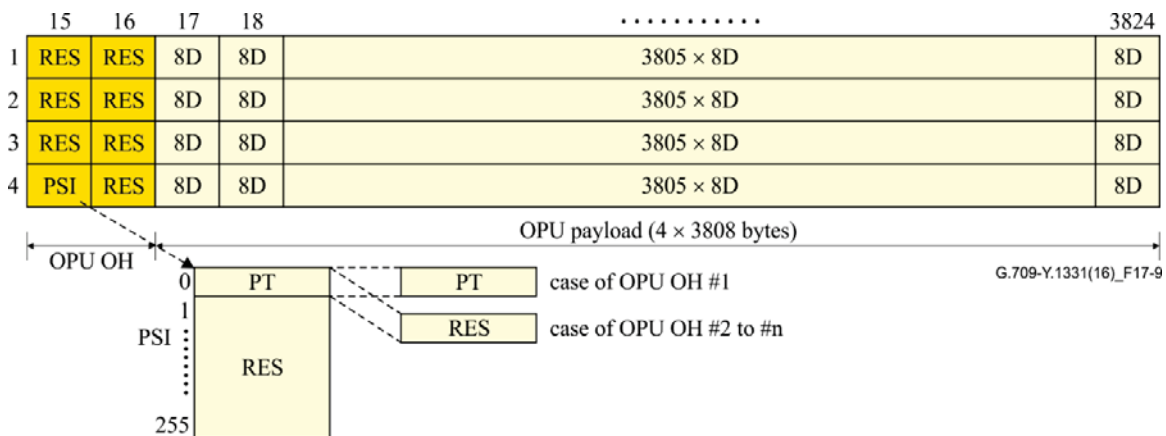


Figure 17-9 – OPU frame structure and mapping of 2 147 483 647-bit pseudo-random test sequence into OPU

The OPU overhead for the PRBS mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes reserved for future international standardization (RES).

The OPU payload for the PRBS mapping consists of 4×3808 bytes.

The OPU_k contains one instance of the 2 147 483 647-bit pseudo-random test sequence. The OPU_{Cn} contains n instances of the 2 147 483 647-bit pseudo-random test sequence, numbered 1 to n.

NOTE – This PRBS test pattern is not intended to be deployed for stress testing of the physical interface.

17.6 Mapping of a non-specific client bit stream into OPU_k

In addition to the mappings of specific client signals as specified in the other subclauses of this clause, a non-specific client mapping into OPU_k is specified. Any (set of) client signal(s), which after encapsulation into a continuous bit stream with a bit rate of the OPU_k payload, can be mapped into the OPU_k payload (see Figure 17-10). The bit stream must be synchronous with the OPU_k signal. Any justification must be included in the continuous bit stream creation process. The continuous bit stream must be scrambled before mapping into the OPU_k payload.

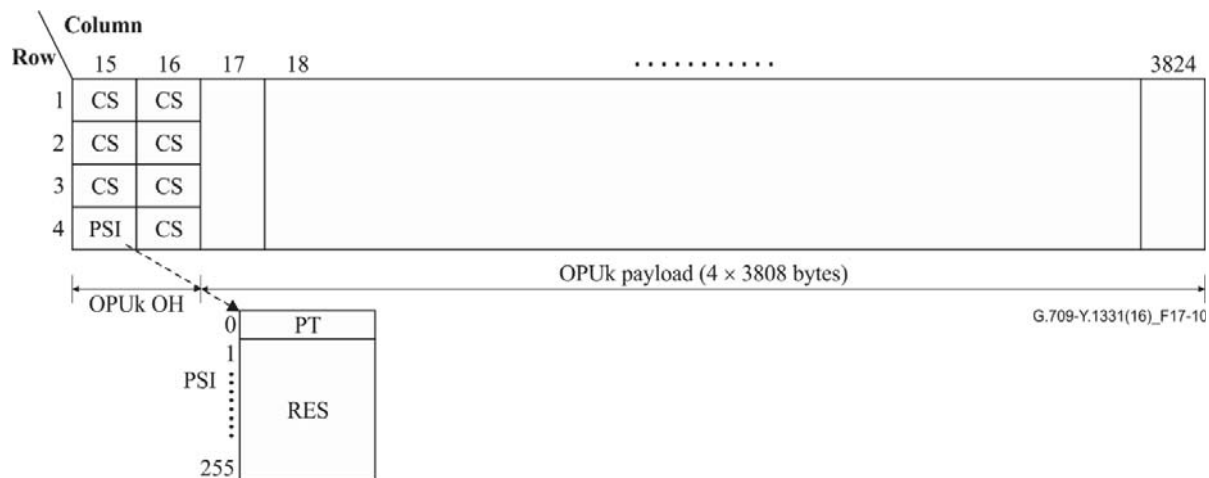


Figure 17-10 – OPUk frame structure for the mapping of a synchronous constant bit stream

The OPUk overhead for the mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), and seven bytes for client-specific (CS) purposes. The definition of these CS overhead bytes is performed within the encapsulation process specification.

The OPUk payload for this non-specific mapping consists of 4×3808 bytes.

17.6.1 Mapping bit stream with octet timing into OPUk

If octet timing is available, each octet of the incoming data stream will be mapped into a data byte (octet) of the OPUk payload.

17.6.2 Mapping bit stream without octet timing into OPUk

If octet timing is not available, groups of eight successive bits (not necessarily an octet) of the incoming data stream will be mapped into a data byte (octet) of the OPUk payload.

17.7 Mapping of other constant bit-rate signals with justification into OPUk

Mapping of other CBR client signals (with up to ± 100 ppm bit-rate tolerance) into an OPUk ($k = 0, 1, 2, 3, 4$) is performed by the generic mapping procedure as specified in Annex D.

During a signal fail condition of the incoming CBR client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the appropriate replacement signal as defined in the clauses hereafter.

During a signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition), the failed client signal is replaced by the appropriate replacement signal as defined in the clauses hereafter.

The OPUk overhead for this mapping consists of a:

- payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-9, the client signal fail (CSF) and 254 bytes plus 7 bits reserved for future international standardization (RES);
- three justification control (JC1, JC2, JC3) bytes carrying the value of GMP overhead C_m ;
- three justification control (JC4, JC5, JC6) bytes carrying the value of GMP overhead ΣC_{nD} and
- one byte reserved for future international standardization (RES).

The JC1, JC2 and JC3 bytes consist of a 14-bit C_m field (bits C1, C2, ..., C14), a 1-bit Increment Indicator (II) field, a 1-bit Decrement Indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The JC4, JC5 and JC6 bytes consist of a 10-bit ΣC_{nD} field (bits D1, D2, ..., D10), a 5-bit CRC-5 field which contains an error check code over the bits 4 to 8 in the JC4, JC5 and JC6 fields and nine bits reserved for future international standardization (RES). The default value of n in ΣC_{nD} is 8. The support for $n=1$ is client dependent and specified in the clauses hereafter when required.

The values of m , $C_{m,min}$, $C_{m,max}$, n , $C_{n,min}$ and $C_{n,max}$ for CBR client into OPUk are as follows:

$$m = 8,16,64,256,640 \quad (17-1)$$

$$c_{m,nom} = \left(\frac{CBR_nom_client_bit_rate \times Number_of_GMP_blocks_in_OPUk}{OPUk_nom_bit_rate} \right) \quad (17-2)$$

$$c_{m,min} = c_{m,nom} \times \left(\frac{1 - CBR_client_bit_rate_tolerance}{1 + OPUk_bit_rate_tolerance} \right) \quad (17-3)$$

$$c_{m,max} = c_{m,nom} \times \left(\frac{1 + CBR_client_bit_rate_tolerance}{1 - OPUk_bit_rate_tolerance} \right) \quad (17-4)$$

$$C_{m,min} = floor(c_{m,min}) \quad (17-5)$$

$$C_{m,max} = ceiling(c_{m,min}) \quad (17-6)$$

$$n = 8,1 \quad (17-7)$$

$$c_{n,nom} = \left(\frac{CBR_client_nom_bit_rate \times Number_of_GMP_blocks_in_OPUk}{OPUk_nom_bit_rate} \right) \quad (17-8)$$

$$c_{n,min} = c_{n,nom} \times \left(\frac{1 - CBR_client_bit_rate_tolerance}{1 + OPUk_bit_rate_tolerance} \right) \quad (17-9)$$

$$c_{n,max} = c_{n,nom} \times \left(\frac{1 + CBR_client_bit_rate_tolerance}{1 - OPUk_bit_rate_tolerance} \right) \quad (17-10)$$

$$C_{n,min} = floor(c_{n,min}) \quad (17-11)$$

$$C_{n,max} = ceiling(c_{n,min}) \quad (17-12)$$

$C_{m,min}$, $C_{n,min}$, $C_{m,max}$ and $C_{n,max}$ values represent the boundaries of client/OPU ppm offset combinations (i.e., min. client/max. OPU and max. client/min. OPU). In steady state, given instances of client/OPU offset combinations should not result in generated C_m and C_n values throughout this range but rather should be within as small a range as possible.

Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

17.7.1 Mapping a sub-1.238 Gbit/s CBR client signal into OPU0

Table 17-4 specifies the clients defined by this Recommendation and their GMP m_n and C_{nD} parameter values. Table 17-5 specifies the replacement signals for those clients.

The support for 1-bit timing information (C_1) is client dependent. Clients for which the 8-bit timing information in C_m with $m=8$ is sufficient will not deploy the ability to transport ΣC_{ID} and the JC4/5/6 value will be fixed to all-0s.

The OPU0 payload for this mapping consists of 4×3808 bytes. The bytes in the OPU0 payload area are numbered from 1 to 15232. The OPU0 payload byte numbering for GMP 1-byte (8-bit) blocks is illustrated in Figure 17-11. In row 1 of the OPU0 frame the first byte will be labelled 1, the next byte will be labelled 2, etc.

Groups of eight successive bits (not necessary being a byte) of the client signal are mapped into a byte of the OPU0 payload area under control of the GMP data/stuff control mechanism. Each byte in the OPU0 payload area may either carry 8 client bits, or carry 8 stuff bits. The stuff bits are set to zero.

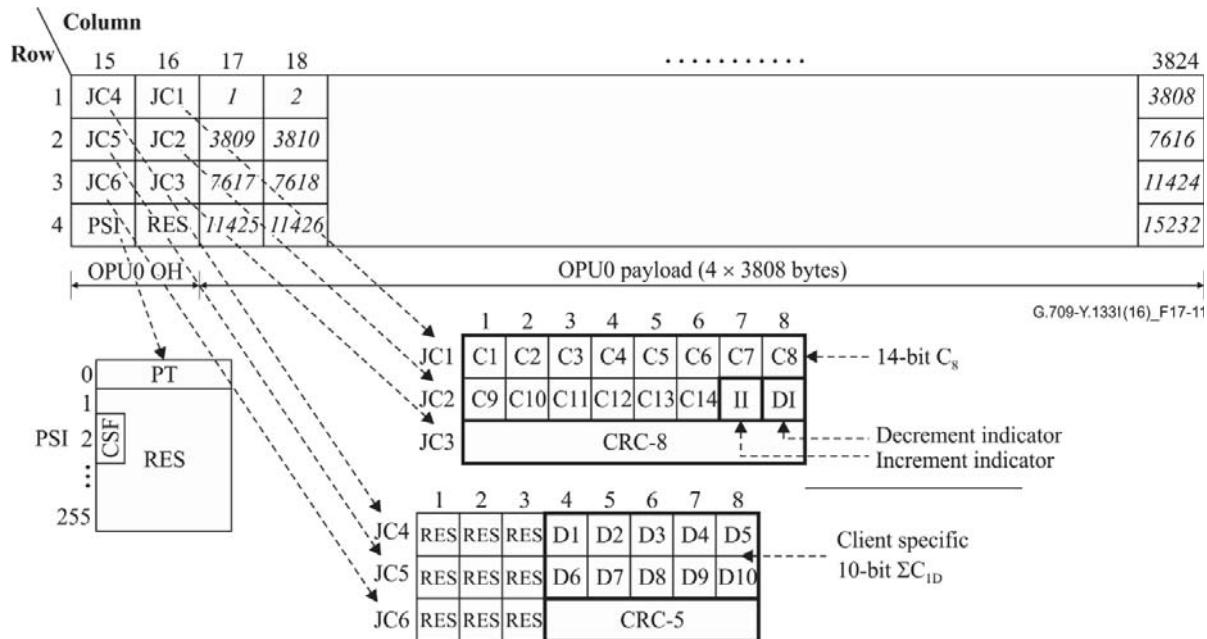


Figure 17-11 – OPU0 frame structure for the mapping of a sub-1.238 Gbit/s client signal

Table 17-4 – m , n and C_{nD} for sub-1.238G clients into OPU0

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	m	n	C_{nD}
Transcoded 1000BASE-X (see clause 17.7.1.1)	$15/16 \times 1\,250\,000$	± 100	8	8	No
STM-1	155 520	± 20	8	1	Yes
STM-4	622 080	± 20	8	1	Yes
FC-100	1 062 500	± 100	8	8	No
SBCON/ESCON	200 000	± 200	8	8	No
DVB-ASI	270 000	± 100	8	8	No
SDI	270 000	± 2.8	8	TBD	TBD

Table 17-5 – Replacement signal for sub-1.238G clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
STM-1	Generic-AIS	± 20
STM-4	Generic-AIS	± 20
1000BASE-X	Link Fault	± 100
FC-100	NOS	± 100
SBCON/ESCON	NOS	± 200
DVB-ASI	Generic-AIS	± 100
SDI	Generic-AIS	For further study

17.7.1.1 1000BASE-X transcoding

The 1000BASE-X signal (8B/10B coded, nominal bit rate of 1 250 000 kbit/s and a bit-rate tolerance up to ± 100 ppm) is synchronously mapped into a 75-octet GFP-T frame stream with a bit rate of $15/16 \times 1\,250\,000$ kbit/s ± 100 ppm (approximately 1 171 875 kbit/s ± 100 ppm). This process is referred to as "timing transparent transcoding (TTT)". The $15/16 \times 1\,250\,000$ kbit/s ± 100 ppm signal is then mapped into an OPU0 by means of the generic mapping procedure as specified in clause 17.7.1 and Annex D.

For 1000BASE-X client mapping, 1-bit timing information (C_1) is not needed, so OPU0 JC4/JC5/JC6 OH value will be fixed to all-0s.

The mapping of the 1000BASE-X signal into GFP-T is performed as specified in [ITU-T G.7041] with the following parameters:

- Each GFP-T frame contains one superblock
- The 65B_PAD character is not used
- GFP idle frames are not used
- The GFP frame pFCS is not used.

During a signal fail condition of the incoming 1000BASE-X client signal (e.g., in the case of a loss of input signal), either:

- This failed incoming 1000BASE-X signal is replaced by a stream of 10B blocks, with a bit rate of $1\,250\,000$ kbit/s ± 100 ppm, each carrying a link fault indication as specified in [IEEE 802.3], which stream is then applied at the GFP-T mapper, or
- The GFP-T signal is replaced by a stream of GFP client signal fail (CSF) and GFP-idle frames as specified in [ITU-T G.7041] with a bit rate of $15/16 \times 1\,250\,000$ kbit/s ± 100 ppm.

During either

- A signal fail condition of the incoming ODU0/OPU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), or
- Incoming CSF frames as specified in [ITU-T G.7041].

The GFP-T de-mapper process generates a stream of 10B blocks, with each block carrying a link fault indication as specified in [IEEE 802.3] as a replacement signal for the lost 1000BASE-X signal.

NOTE – The Ethernet link fault indication is a stream of repeating /C1/C2/C1/C2/ ... ordered sets, where $C1 = /K28.5/D21.5/D0.0/D0.0/$ and $C2 = /K28.5/D2.2/D0.0/D0.0/$. This character stream is then processed by the GFP-T mapper process in the same manner as if it were the received 8B/10B data stream, mapping it into GFP-T superblocks for transmission.

17.7.1.2 FC-100

During a signal fail condition of the incoming FC-100 signal (e.g., in the case of a loss of input signal), this failed incoming FC-100 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-100 signal.

17.7.1.3 SBCON/ESCON

During a signal fail condition of the incoming SBCON/ESCON signal (e.g., in the case of a loss of input signal), this failed incoming SBCON/ESCON signal is replaced by an NOS sequence as specified in [b-ANSI INCITS 296].

NOTE – The NOS sequence ordered set is defined as /K28.5/D0.2/.

During a signal fail condition of the incoming ODU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), NOS sequence ordered sets as specified in [b-ANSI INCITS 296] are generated as a replacement signal for the lost SBCON/ESCON signal.

17.7.2 Mapping a supra-1.238 to sub-2.488 Gbit/s CBR client signal into OPU1

Table 17-6 specifies the clients defined by this Recommendation and their GMP m , n and C_{nD} parameter values. Table 17-7 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU1 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU1 JC4/JC5/JC6 OH is client dependent.

The OPU1 payload for this mapping consists of 4×3808 bytes. The groups of 2 bytes in the OPU1 payload area are numbered from 1 to 7616. The OPU1 payload byte numbering for GMP 2-byte (16-bit) blocks is illustrated in Figure 17-12. In row 1 of the OPU1 frame the first 2-bytes will be labelled 1, the next 2-bytes will be labelled 2, etc.

Groups of sixteen successive bits of the client signal are mapped into a group of 2 successive bytes of the OPU1 payload area under control of the GMP data/stuff control mechanism. Each group of 2 bytes in the OPU1 payload area may either carry 16 client bits, or carry 16 stuff bits. The stuff bits are set to zero.

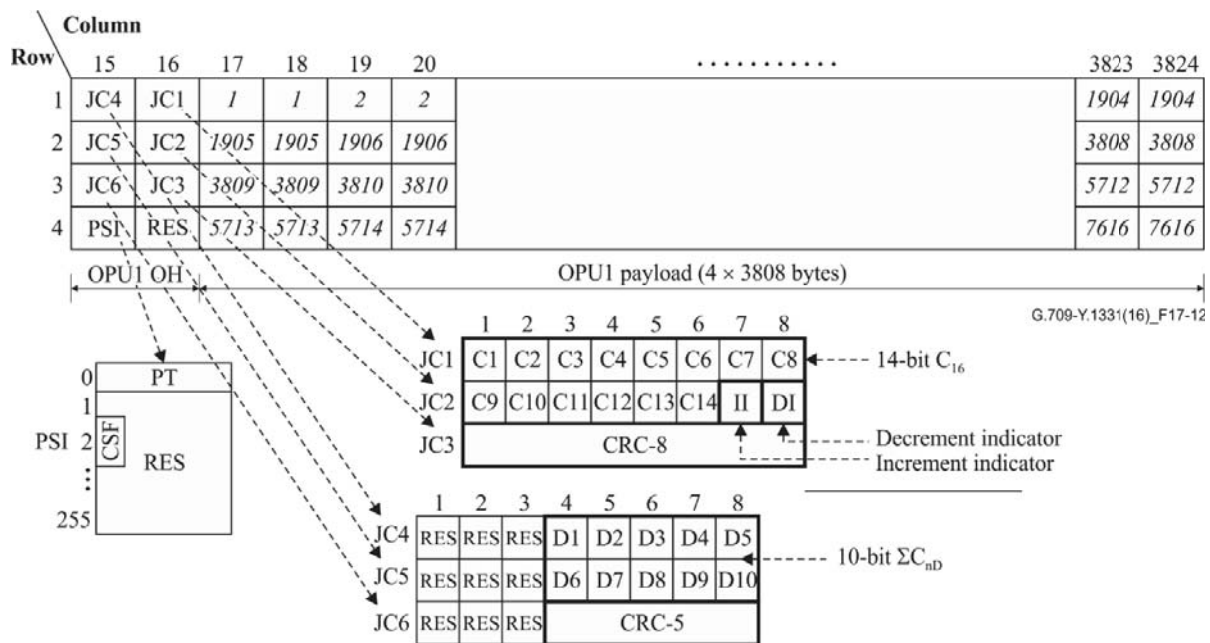


Figure 17-12 – OPU1 frame structure for the mapping of a supra-1.238 to sub-2.488 Gbit/s client signal

Table 17-6 – C_m (m=16) for supra-1.238 to sub-2.488G clients into OPU1

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	m	n	C _{nd}
FC-200	2 125 000	±100	16	8	Yes
1.5G SDI	1 485 000	±10	16	TBD	Yes
1.5G SDI	1 485 000/1.001	±10	16	TBD	Yes

Table 17-7 – Replacement signal for supra-1.238 to sub-2.488 Gbit/s clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
FC-200	NOS	± 100
1.5G SDI	Generic-AIS	For further study

17.7.2.1 FC-200

During a signal fail condition of the incoming FC-200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-200 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODU1 signal (e.g., in the case of an ODU1-AIS, ODU1-LCK, ODU1-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-200 signal.

17.7.3 Mapping CBR client signals into OPU2

Table 17-8 specifies the clients defined by this Recommendation and their GMP m , n and C_{ND} parameter values. Table 17-9 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU2 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU2 JC4/JC5/JC6 OH is client dependent.

The OPU2 payload for this mapping consists of 4×3808 bytes. The groups of eight bytes in the OPU2 payload area are numbered from 1 to 1904. The OPU2 payload byte numbering for GMP 8-byte (64-bit) blocks is illustrated in Figure 17-13. In row 1 of the OPU2 frame the first 8-bytes will be labelled 1, the next 8-bytes will be labelled 2, etc.

Groups of sixty-four successive bits of the client signal are mapped into a group of eight successive bytes of the OPU2 payload area under control of the GMP data/stuff control mechanism. Each group of eight bytes in the OPU2 payload area may either carry 64 client bits, or carry 64 stuff bits. The stuff bits are set to zero.

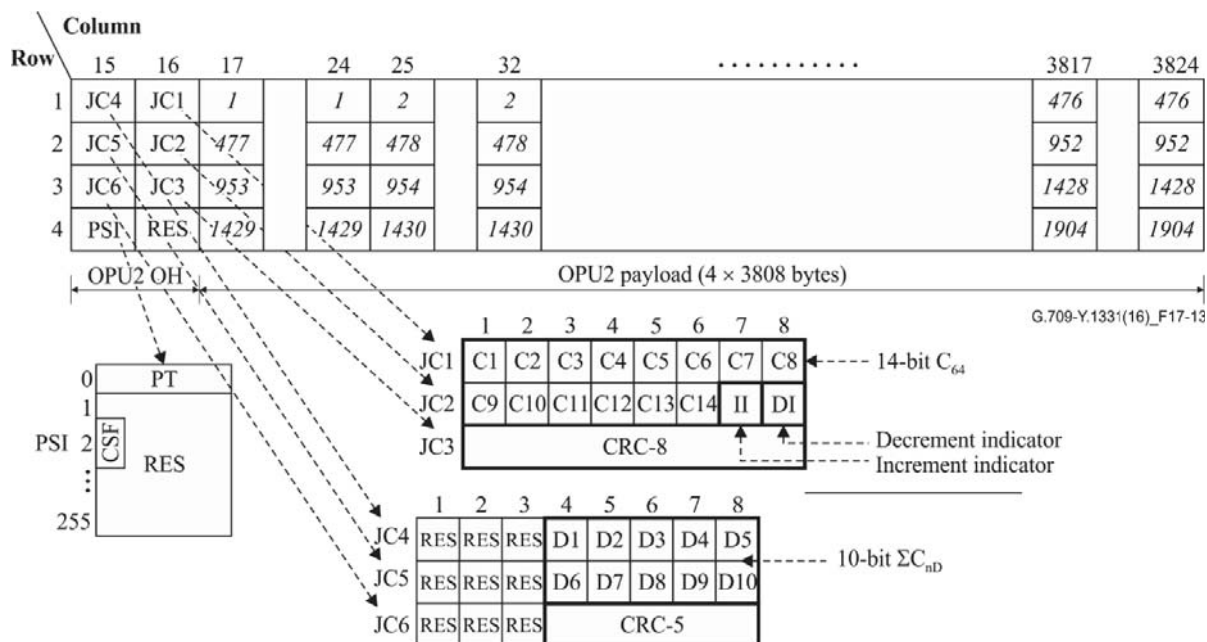


Figure 17-13 – OPU2 frame structure for the mapping of a CBR client signal

Table 17-8 – m, n and C_{ND} for CBR clients into OPU2

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	m	n	C _{nD}
For further study					

Table 17-9 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
For further study		

17.7.4 Mapping CBR client signals into OPU3

Table 17-10 specifies the clients defined by this Recommendation and their GMP m , n and C_{nD} parameter values. Table 17-11 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU3 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU3 JC4/JC5/JC6 OH is client dependent.

The OPU3 payload for this mapping consists of 4×3808 bytes. The groups of 32 bytes in the OPU3 payload area are numbered from 1 to 476. The OPU3 payload byte numbering for GMP 32-byte (256-bit) blocks is illustrated in Figure 17-14. In row 1 of the OPU3 frame the first 32-bytes will be labelled 1, the next 32-bytes will be labelled 2, etc.

Groups of two hundred-fifty-six successive bits of the client signal are mapped into a group of 32 successive bytes of the OPU3 payload area under control of the GMP data/stuff control mechanism. Each group of 32 bytes in the OPU3 payload area may either carry 256 client bits, or carry 256 stuff bits. The stuff bits are set to zero.

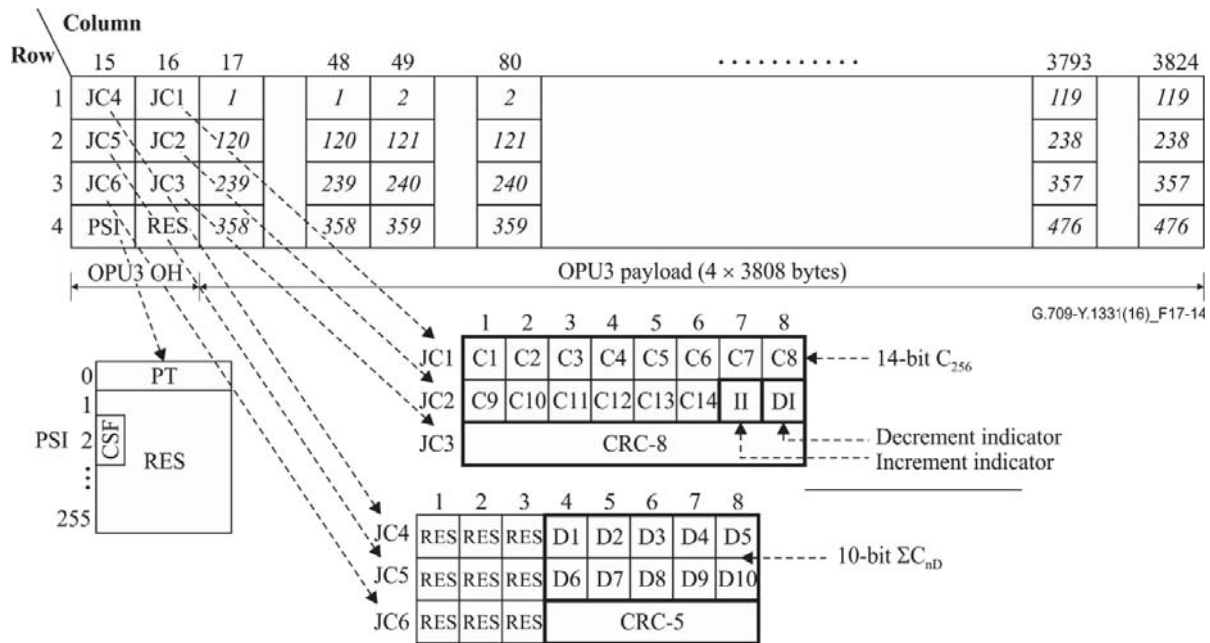


Figure 17-14 – OPU3 frame structure for the mapping of a CBR client signal

Table 17-10 – m, n and C_{nD} for CBR clients into OPU3

Client signal	Nominal bit rate (kbit/s)	Bit rate tolerance (ppm)	m	n	C_{nD}
Transcoded 40GBASE-R (see clause 17.7.4.1)	$1027/1024 \times 64/66 \times 41\,250\,000$	± 100	256	8	Yes

Table 17-11 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
40GBASE-R	Continuous 40GBASE-R local fault sequence ordered sets with four PCS lane alignment markers inserted after each 16383×4 sixty-six-bit blocks	± 100

A 40GBASE-R local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

17.7.4.1 40GBASE-R multi-lane processing and transcoding

The 40GBASE-R client signal (64B/66B encoded, nominal aggregate bit-rate of 41 250 000 kbit/s, ± 100 ppm) is recovered using the process described in Annex E for parallel 64B/66B interfaces. The lane(s) of the physical interface are bit-disinterleaved, if necessary, into four streams of 10 312 500 kbit/s. 66B block lock and lane alignment marker lock are acquired on each PCS lane, allowing the 66B blocks to be de-skewed and reordered.

The resulting sequence is descrambled and transcoded according to the process described in Annex B into 513B code blocks. Each pair of two 513B code blocks is combined according to the process described in Annex F into a 1027B block, resulting in a bit stream of $1027/1024 \times 40\,000\,000$ kbit/s ± 100 ppm (40,117,187.500 kbit/s ± 100 ppm). This process is referred to as "timing transparent transcoding (TTT)", mapping a bit stream which is 1027/1056 times the bit-rate of the aggregate Ethernet signal.

In the mapper, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

The de-mapper will insert a compensated Ethernet PCS lane BIP as described in Annex E. In addition, as described in Annex E, the combined error mask resulting from the PCS BIP-8 error mask and the OTN BIP-8 error mask may be used as a non-intrusive monitor.

For 40GBASE-R client mapping, 1-bit timing information (C_1) is not needed.

The de-mapper will recover from the output of the GMP processor 1027B block lock, and then trans-decode each 1027B block to sixteen 66B blocks as described in Annex E. Trans-decoded lane alignment markers are constructed with a compensated BIP-8. The 66B blocks are then re-distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

17.7.5 Mapping CBR client signals into OPU4

Table 17-12 specifies the clients defined by this Recommendation and their GMP m, n and C_{nD} parameter values. Table 17-13 specifies the replacement signals for those clients.

The support for 8-bit timing information (ΣC_{8D}) in the OPU4 JC4/JC5/JC6 OH is required.

The support for 1-bit timing information (ΣC_{1D}) in the OPU4 JC4/JC5/JC6 OH is client dependent.

The OPU4 payload for this mapping consists of 4×3800 bytes for client data and 4×8 bytes with fixed stuff. The groups of 80 bytes in the OPU4 payload area are numbered from 1 to 190. The OPU4 payload byte numbering for GMP 80-byte (640-bit) blocks is illustrated in Figure 17-15. In row 1 of the OPU4 frame the first 80-bytes will be labelled 1, the next 80-bytes will be labelled 2, etc.

Groups of six hundred and forty successive bits of the client signal are mapped into a group of 80 successive bytes of the OPU4 payload area under control of the GMP data/stuff control mechanism. Each group of 80 bytes in the OPU4 payload area may either carry 640 client bits, or carry 640 stuff bits. The stuff bits are set to zero.

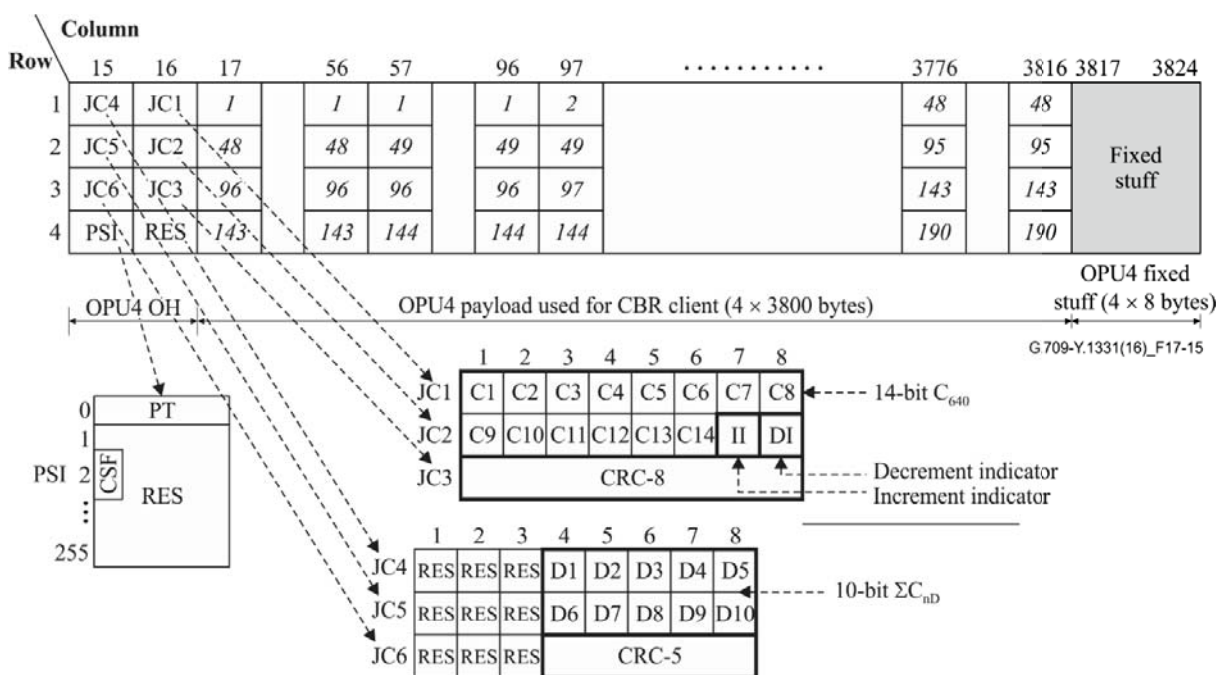


Figure 17-15 – OPU4 frame structure for the mapping of a CBR client signal

Table 17-12 – m, n and C_{nD} for CBR clients into OPU4

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)	m	n	C _{nD}
100GBASE-R (see 17.7.5.1)	103 125 000	±100	640	8	Yes

Table 17-13 – Replacement signal for CBR clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
100GBASE-R (see 17.7.5.1)	Continuous 100GBASE-R local fault sequence ordered sets with 20 PCS lane alignment markers inserted after each 16383 x 20 sixty-six-bit blocks	±100

A 100GBASE-R local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate a "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

17.7.5.1 100GBASE-R multi-lane processing

The 100GBASE-R client signal (64B/66B encoded, nominal aggregate bit-rate of 103 125 000 kbit/s ± 100 ppm) is recovered using the process described in Annex E for parallel 64B/66B interfaces. The lane(s) of the physical interface are bit-disinterleaved, if necessary, into twenty streams of 5 161 250 kbit/s. 66B block lock and lane alignment marker lock are acquired on each PCS lane, allowing the 66B blocks to be de-skewed and reordered.

In the mapper, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

The de-mapper will pass through the PCS lane BIP from the ingress as described in Annex E. In addition, the received Ethernet PCS lane BIP may be compared with the expected Ethernet PCS lane BIP as a non-intrusive monitor.

For 100GBASE-R client mapping, 1-bit timing information (C_1) is not needed.

The de-mapper will recover from the output of the GMP processor 64B/66B block lock per the state diagram in Figure 82-10 [IEEE 802.3]. The 66B blocks are re-distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

17.8 Mapping a 1000BASE-X and FC-1200 signal via timing transparent transcoding into OPU_k

17.8.1 Mapping a 1000BASE-X signal into OPU₀

Refer to clause 17.7.1 for the mapping of the transcoded 1000BASE-X signal and to clause 17.7.1.1 for the transcoding of the 1000BASE-X signal.

17.8.2 Mapping an FC-1200 signal into OPU_{2e}

The nominal line rate for FC-1200 is 10 518 750 kbit/s \pm 100 ppm, and must therefore be compressed to a suitable rate to fit into an OPU_{2e}.

The adaptation of the 64B/66B encoded FC-1200 client is done by transcoding a group of eight 66B blocks into one 513B block (as described in Annex B), assembling eight 513B blocks into one 516-octet superblock and encapsulating seventeen 516-octet superblocks into an 8800 octet GFP frame as illustrated in Figure 17-17. The GFP frame consists of 2200 rows with 32 bits per row. The first row contains the GFP core header, the second row the GFP payload header. The next four rows contain 16 bytes reserved for future international standardization. The next seventeen times 129 rows contain the seventeen superblocks #1 to #17. The last row contains the GFP payload FCS. The flag (F) bit of 513B block # i ($i = 0..7$) is carried in Flag # i bit located in the superblock flags field. The remaining 512 bits of each of the eight 513B blocks of a superblock are carried in 16 rows of the superblock data field; bits of 513B block #0 in the first 16 rows of the superblock, bits of 513B block #1 in the next 16 rows, etc. Each 513B block contains 'j' ($j = 0..8$) control blocks (CB1 to CB j) and '8-j' all-data blocks (DB1..DB8-j) as specified in Annex B. Figure 17-17 presents a 513B block with three control blocks and five all-data blocks. A 513B block may contain zero to eight control blocks and a superblock may contain thus zero to sixty-four control blocks.

NOTE 1 – The GFP encapsulation stage does not generate GFP-idle frames and therefore the generated GFP stream is synchronous to the FC-1200 client stream. The adaptation process performs a 50/51 rate compression, so the resulting GFP stream has a signal bit rate of $50/51 \times 10.51875$ Gbit/s \pm 100 ppm (i.e., 10 312 500 kbit/s \pm 100 ppm).

The stream of 8800 octet GFP frames is byte-synchronous mapped into the OPU_{2e} payload by aligning the byte structure of every GFP frame with the byte structure of the OPU_{2e} payload (see Figure 17-16). Sixty-four fixed stuff (FS) bytes are added in columns 1905 to 1920 of the OPU_{2e} payload. All the GFP frames have the same length (8800 octets). The GFP frames are not aligned with the OPU_{2e} payload structure and may cross the boundary between two OPU_{2e} frames.

During a signal fail condition of the incoming FC-1200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-1200 signal is replaced by a stream of 66B blocks, with each block carrying two local fault sequence ordered sets as specified in [b-ANSI INCITS 364]. This replacement signal is then applied at the transcoding process.

NOTE 2 – Local fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH=10 0x55 00 00 01 00 00 00 01.

During a signal fail condition of the incoming ODU_{2e}/OPU_{2e} signal (e.g., in the case of an ODU_{2e}-AIS, ODU_{2e}-LCK, ODU_{2e}-OCI condition) a stream of 66B blocks, with each block carrying

two local fault sequence ordered sets as specified in [b-ANSI INCITS 364] is generated as a replacement signal for the lost FC-1200 signal.

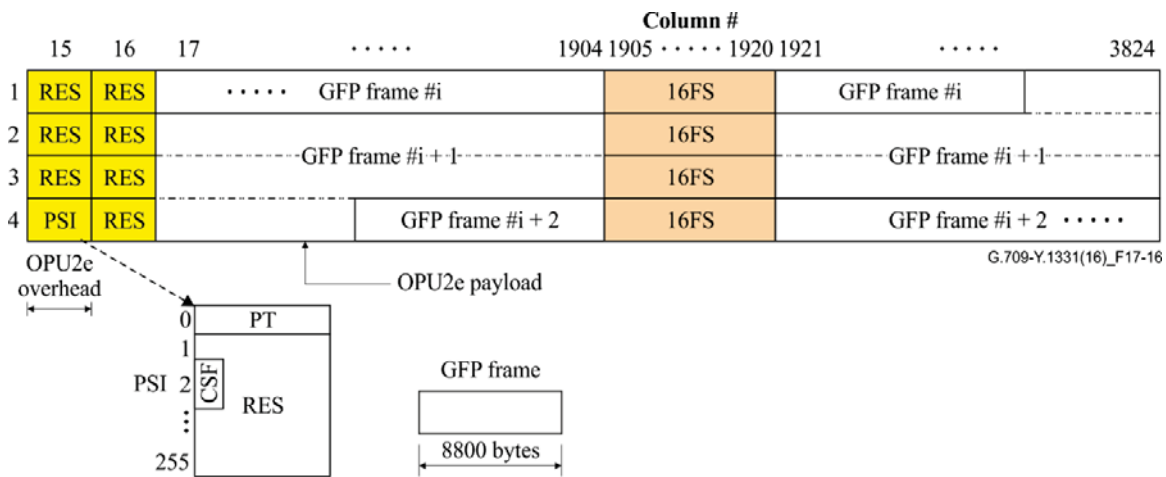


Figure 17-16 – Mapping of transcoded FC-1200 into OPU2e

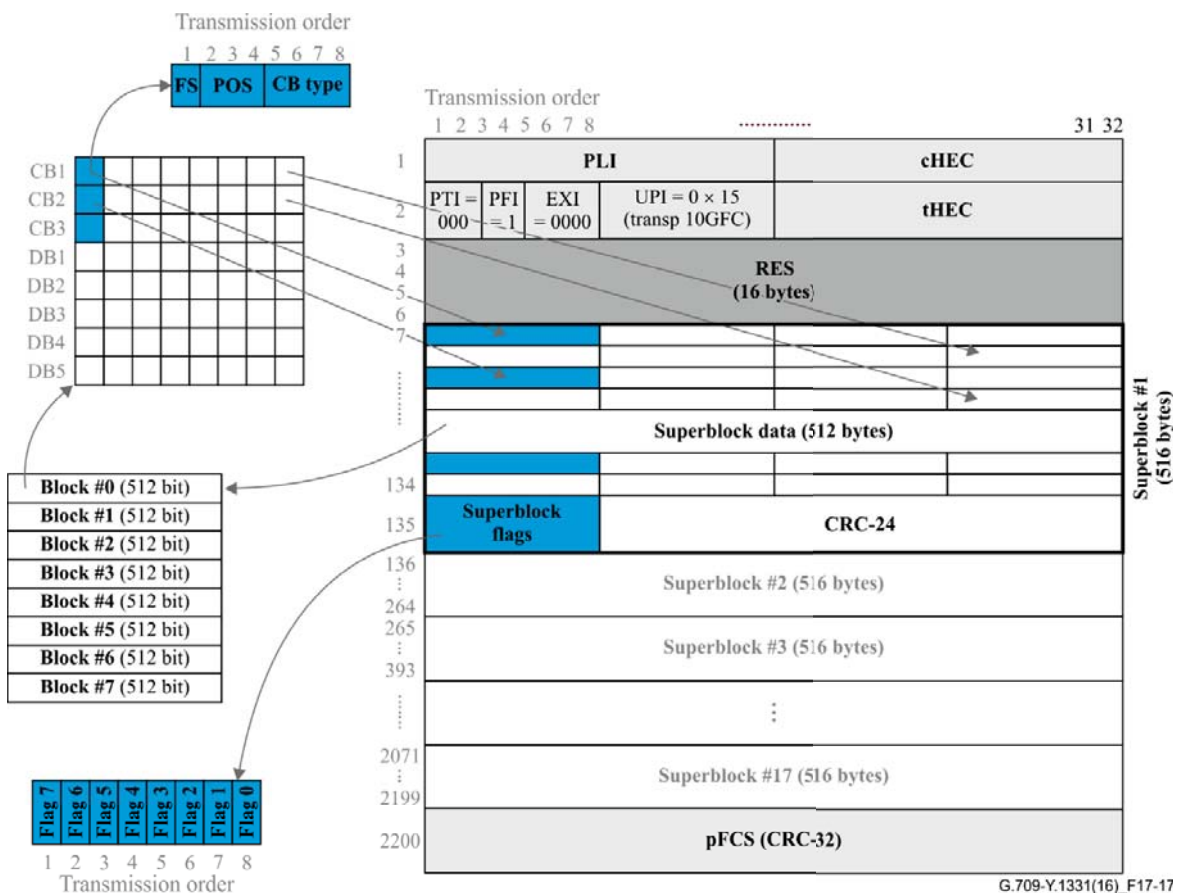


Figure 17-17 – GFP frame format for FC-1200

GFP framing is used to facilitate delineation of the superblock structure by the receiver. The leading flag bits from each of the eight 513B blocks are relocated into a single octet at the end of the 513-octet superblock data field (labelled "Superblock flags").

To minimize the risk of incorrect decoding due to errors in the 1 to 65 octets of "control" information (Flags, FC, POS, CB_Type), a CRC-24 is calculated over the 65 octets within each superblock that may contain such "control" information and appended to form a 516 octet superblock. The 65 octets in the 516-octet superblock over which the CRC-24 is calculated are the octets $(1+8n)$ with $n=0..64$ (i.e., octets 1, 9, 17, ..., 513). The generator polynomial for the CRC-24 is $G(x) = x^{24} + x^{21} + x^{20} + x^{17} + x^{15} + x^{11} + x^9 + x^8 + x^6 + x^5 + x + 1$ with an all-ones initialization value, where x^{24} corresponds to the MSB and x^0 to the LSB. This superblock CRC is generated by the source adaptation process using the following steps:

- 1) The 65 octets of "control" information (Flags, POS, CB_Type) are taken in network octet order (see Figure 17-17), most significant bit first, to form a 520-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 519.
- 2) $M(x)$ is multiplied by x^{24} and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 23 or less.
- 3) The coefficients of $R(x)$ are considered to be a 24-bit sequence, where x^{23} is the most significant bit.
- 4) After inversion, this 24-bit sequence is the CRC-24.

Exactly 17 of these 516-octet superblocks are prefixed with the standard GFP core and type headers and 16 octets of "reserved" (padding). Because the number of 516-octet superblocks per GFP frame is known a priori, it is possible for this mapping scheme to operate in a cut-through (as opposed to store and forward) fashion, thus minimizing the mapping latency.

The payload FCS (a CRC-32) is appended to the end of each GFP frame and is calculated across the payload information field of the GFP frame as per [ITU-T G.7041]. The purpose of the payload FCS is to provide visibility of bit errors occurring anywhere in the GFP payload information field and thus augments the coverage provided by the per-superblock CRC-24 (which only provides coverage for the "control" overhead in each superblock). The payload FCS is only for the purposes of gathering statistics.

All octets in the GFP payload area are scrambled using the $X^{43} + 1$ self-synchronous scrambler, again as per [ITU-T G.7041].

17.9 Mapping a supra-2.488 Gbit/s CBR signal into OPUflex using BMP

Mapping of a supra-2.488 Gbit/s CBR client signal (with up to ± 100 ppm bit-rate tolerance) into an OPUflex is performed by a bit-synchronous mapping procedure (BMP). Table 17-14 specifies the clients defined by this Recommendation.

The bit-synchronous mapping process deployed to map constant bit rate client signals into an OPUflex does not generate any justification control signals.

The OPUflex clock for the bit-synchronous mapping is derived from the client signal. During a signal fail condition of the incoming client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the appropriate replacement signal as defined in Table 17-15. The OPUflex payload signal bit rate shall be within the limits specified in Table 7-3 and neither an OPUflex frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming client signal shall be done without introducing an OPUflex frequency or frame phase discontinuity.

During a signal fail condition of the incoming ODUflex/OPUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), the failed client signal is replaced by the appropriate replacement signal as defined in Table 17-15.

The OPUflex overhead for this mapping consists of:

- A payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-9, the client signal fail (CSF) and 254 bytes plus 7 bits reserved for future international standardization (RES);
- Three justification control (JC) bytes, consisting of two bits for justification control (with fixed 00 value) and six bits reserved for future international standardization;
- One negative justification opportunity (NJO) byte (carrying a justification byte); and
- Three bytes reserved for future international standardization (RES).

NOTE – To allow the use of a common asynchronous/bit-synchronous de-mapper circuit for CBR client signals into ODUk (k=1,2,3 and flex), JC, NJO and PJO fields are assumed to be present in the OPUflex frame structure for the mapping of a supra-2.488G CBR client signal (Figure 17-18). This OPUflex frame structure is now compatible with the OPUk frame structure for the mapping of a CBR2G5, CBR10G or CBR40G signal (Figure 17-1). As a CBR signal is mapped into the OPUflex without justification, the NJO field contains a justification byte (stuff), the PJO field contains a data byte (D), and the JC bits are fixed to 00.

The OPUflex payload for this mapping consists of 4×3808 bytes (Figure 17-18). Groups of eight successive bits (not necessarily being a byte) of the client signal are mapped into a data (D) byte of the OPUflex payload area under control of the BMP control mechanism. Each data byte in the OPUflex payload area carries 8 client bits.

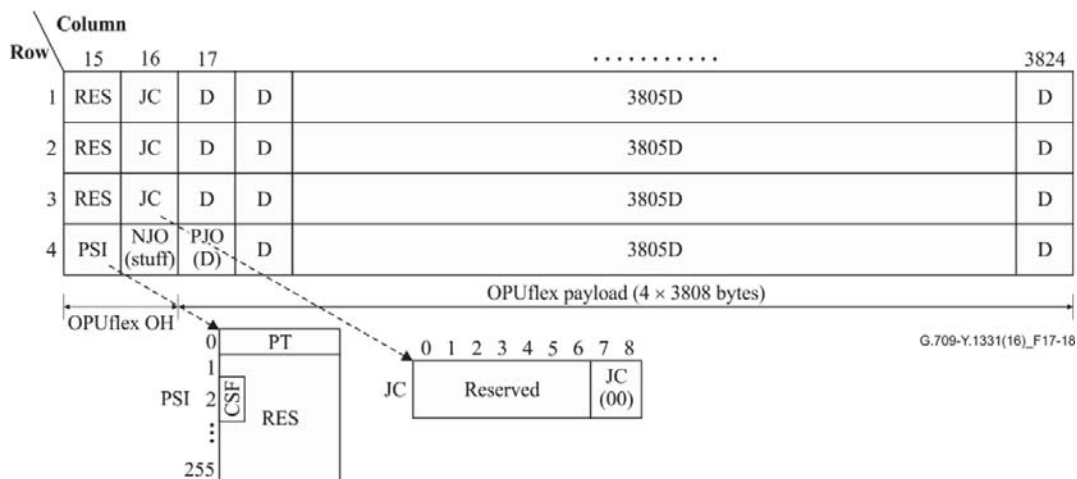


Figure 17-18 – OPUflex frame structure for the mapping of a supra-2.488 Gbit/s client signal

Table 17-14 – supra-2.488G CBR clients

Client signal	Nominal bit rate (kbit/s)	Bit-rate tolerance (ppm)
FC-400	4 250 000	±100
FC-800	8 500 000	±100
FC-1600	14 025 000	±100
FC-3200	28 050 000	±100
IB SDR	2 500 000	±100
IB DDR	5 000 000	±100
IB QDR	10 000 000	±100
3G SDI	2 970 000	± 10
3G SDI	2 970 000/1.001	±10

Table 17-15 – Replacement signal for supra-2.488 Gbit/s clients

Client signal	Replacement signal	Bit-rate tolerance (ppm)
FC-400	NOS	± 100
FC-800	NOS	± 100
FC-1600	NOS	± 100
FC-3200	NOS	± 100
IB SDR	For further study	± 100
IB DDR	For further study	± 100
IB QDR	For further study	± 100
3G SDI	Generic-AIS	For further study

17.9.1 FC-400 and FC-800

During a signal fail condition of the incoming FC-400/FC-800 signal (e.g., in the case of a loss of input signal), this failed incoming FC-400/FC-800 signal is replaced by an NOS primitive sequence as specified in [b-INCITS 470].

NOTE – The NOS primitive sequence ordered set is defined as /K28.5/D21.2/D31.5/D5.2/.

During a signal fail condition of the incoming ODUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-400/FC-800 signal.

17.9.2 FC-1600

The characteristic information of the mapped FC-1600 client signal consists of a sequence of 64B/66B encoded blocks with a nominal bit-rate of 14 025 000 kbit/s, ± 100 ppm.

In case the FC-1600 interface at the mapper has FEC enabled the mapper must recover the FEC code word synchronization, extract the FEC parity bits, perform error correction and transdecode the 64B/65B blocks to 64B/66B blocks as specified in [b-INCITS 470].

In case the FC-1600 interface at the demapper has FEC enabled the demapper must recover 66B block lock from the demapped CBR signal, transcode the 64B/66B blocks to 64/65B blocks, generate and insert the FEC parity bits as specified in [b-INCITS 470].

NOTE – FC-1600 interface ports designed prior to Edition 4.6 may not be able to support termination of the FEC or transdecoding of 64B/65B blocks.

During a signal fail condition of the incoming FC-1600 signal (e.g., in the case of a loss of input signal), this failed incoming FC-1600 signal is replaced by a NOS primitive sequence as specified in [b-INCITS 470].

During signal fail condition of the incoming ODUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 470] are generated as a replacement signal for the lost FC-1600 signal.

17.9.3 FC-3200

The characteristic information of the mapped FC-3200 client signal consists of a sequence of 64B/66B encoded blocks with a nominal bit-rate of 28 050 000 kbit/s, ± 100 ppm.

The mapper must recover the FEC code word synchronization, extract the FEC parity bits, perform error correction and transdecode the 256B/257B blocks to 64B/66B blocks as specified in [b-INCITS 488]. Uncorrectable FEC code words shall be replaced with error control blocks at the output of the transdecoder.

The demapper must recover 66B block lock from the demapped CBR signal, transcode the 64B/66B blocks to 256/257B blocks, generate and insert the FEC parity bits as specified in [b-INCITS 488].

During a signal fail condition of the incoming FC-3200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-3200 signal is replaced by a NOS primitive sequence as specified in [b-INCITS 488] at the output of the transdecoder.

During signal fail condition of the incoming ODUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), NOS primitive sequence ordered sets as specified in [b-INCITS 488] are generated as a replacement signal for the lost FC-3200 signal at the input of the transcoder.

17.10 Mapping of packet client signals into OPUk

A set of packet client signals with an aggregated bandwidth of less or equal than 100 Gbit/s is encapsulated into GFP-F as specified in [ITU-T G.7041] and then mapped into an OPUk or OPUflex as specified in 17.4.

A set of packet client signals with an aggregated bandwidth of more than 100 Gbit/s is presented (see Note) as an $n \times 25$ Gbit/s stream of Ethernet MAC frames and interframe gaps and then 64b/66b encoded as specified in [IEEE 802.3] Table 82-5 into a FlexE Client signal [OIF FlexE IA] which is then mapped into an OPUflex as specified in clause 17.11.

A set of packet client signals with an aggregated bandwidth of less or equal than 100 Gbit/s may be presented (see Note) as a 10, 25, 40, 50, 75 or 100 Gbit/s stream of Ethernet MAC frames and interframe gaps and then 64b/66b encoded as specified in [IEEE 802.3] Table 82-5 into a FlexE Client signal [OIF FlexE IA] which is then mapped into an OPUflex as specified in clause 17.11.

NOTE – Non Ethernet packet clients are assumed to be encapsulated into Ethernet MAC frames before they are presented. Encapsulation is outside the scope of this Recommendation. Ethernet MAC framed packet clients are presented directly.

17.11 Mapping of FlexE client signals into OPUflex using IMP

FlexE client signal bit rates are $s \times 5,156,250.000 \text{ kbit/s} \pm 100 \text{ ppm}$, with $s = 2, 8, n \times 5$ ($n \geq 1$). Refer to [OIF FlexE IA].

NOTE – Other FlexE client bit rates are for further study.

The 66b block stream shall be scrambled after rate adaptation and before mapping into the OPUflex. In the reverse operation, following termination of the OPUflex signal, the 66 block stream will be descrambled before being passed to the FlexE Client layer.

A self-synchronizing scrambler with generator polynomial $1 + x^{39} + x^{58}$ shall be used, that is identical to the scrambler specified in clause 49.2.6 [IEEE 802.3]. 66b block stream scrambling is required to provide security against false 66b block delineation (as the two sync header bits bypass the scrambler), the 66b codewords replicating the OTU and ODU frame alignment signal and the 66b codewords combined with the OTU scrambler pattern used for the interface replicating the OTU and ODU frame alignment signal.

Mapping of a FlexE client signal (with up to $\pm 100 \text{ ppm}$ bit-rate tolerance) into an OPUflex is performed by the idle mapping procedure (IMP). The OPUflex payload bit rate is $s \times 5,156,250.000 \text{ kbit/s} \pm 100 \text{ ppm}$, with $s = 2, 8, n \times 5$ ($n \geq 1$). The ODUflex bit rate is $s \times 239/238 \times 5,156,250.000 \text{ kbit/s} \pm 100 \text{ ppm}$.

The Idle mapping procedure deploys a clock rate adaptation scheme based on Idle control character (/I/) insert/delete as per clause 82.2.3.6 of [IEEE 802.3] and/or sequence ordered set (/O/) delete as per clause 82.2.3.9 of [IEEE 802.3].

The OPUflex overhead for this mapping consists of a:

- payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-9, the client signal fail (CSF) and 254 bytes plus 7 bits reserved for future international standardization (RES);
- seven bytes reserved for future international standardization (RES).

The OPUflex payload for this mapping consists of 4×3808 bytes (Figure 17-19). Scrambled 66b blocks of the client signal are mapped into 66-bits of the OPUflex payload area under control of the IMP control mechanism. The 66b blocks must begin at OPUflex payload octet bit positions 1, 3, 5, or 7, keeping the 2-bit alignment of each 66b block through the mapping process.

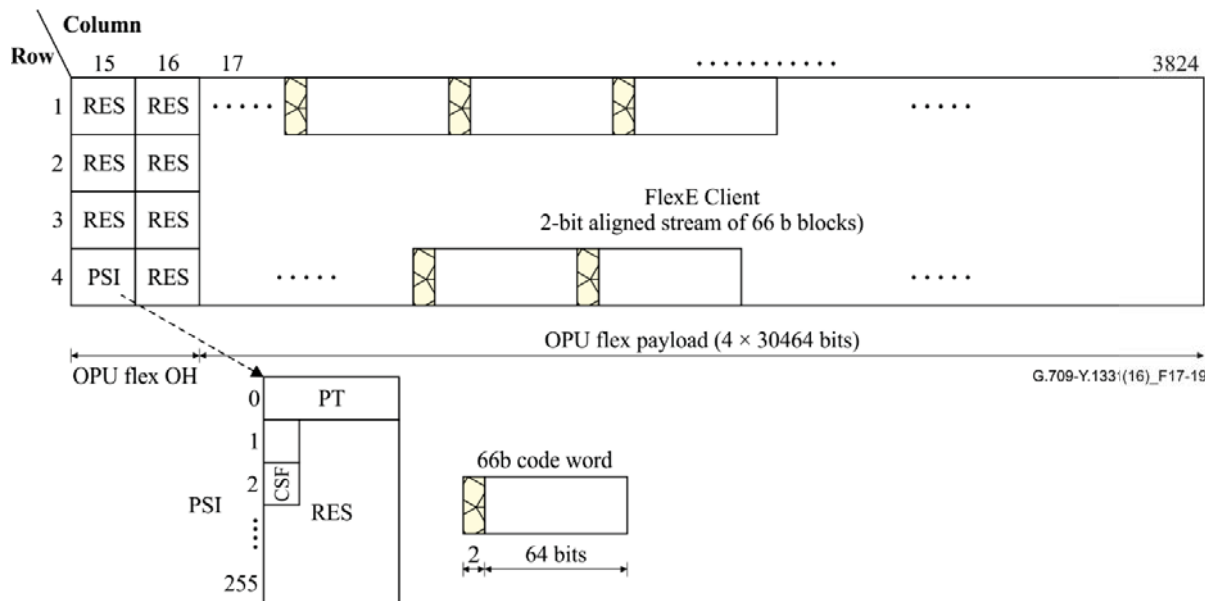


Figure 17-19 – OPUflex frame structure for the mapping of a FlexE client signal

During a signal fail condition of the incoming FlexE client signal (e.g., in the case of a loss of input signal), this failed incoming FlexE client signal is replaced by a stream of 66B blocks, with each block carrying one local fault sequence ordered set (as specified in [FlexE IA]). This replacement signal is then mapped into the OPUflex.

During a signal fail condition of the incoming ODUflex/OPUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition), a stream of 66B blocks, with each block carrying one local fault sequence ordered sets (as specified in [FlexE IA]) is generated as a replacement signal for the lost FlexE client signal.

NOTE 1 – A FlexE client signal local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate a "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

17.12 Mapping of FlexE aware signals into OPUflex

In the FlexE aware service p ($2 \leq p \leq m \leq 252$) 100G FlexE signals – out of an $m \times 100$ G FlexE group signal – are crunched, padded and interleaved as per Figure 17-20. This results in a FlexE partial rate (sub)group signal with a length of $1024 \cdot n$ blocks, with $n = n_1 + n_2 + \dots + n_p$ (FlexEp-n):

- $i = 1 \dots p$ represent the FlexE signals in ascending PHY number order.
- n_i ($i = 1 \dots p$) represents the number of FlexE calendar slots that are to be transferred.

- The value of n_i for each of the p FlexE signals is negotiated between the FlexE partial rate (sub)group mapping port and the FlexE Shim in the customer equipment. q ($0 \leq q \leq p$) out of the p FlexE signals may have their n_i set to 20 while the other $p-p_1$ FlexE signals may have their n_i set to values lower than 20.
- Bits of the Ethernet error control blocks in $20-n_i$ calendar slots that are marked unavailable are located at the end of the sub-calendar and are dropped by the mapper to reduce the rate and reinserted by the demapper to restore the original rate.
- The FlexE partial rate (sub)group interleaving is simplified by adding n_i-1 padding blocks between the overhead block and the first sub-calendar block in each of the p FlexE signals. The value of each padding block is an Ethernet error control block.
- The p OH blocks are interleaved in the order #1, #2 to # p . The p n_i-1 padding blocks are interleaved in the order #1, #2 to # p . The p sub-calendar blocks (of length n_1, n_2 to n_p) are interleaved in the order #1, #2 to # p .

NOTE 1 – An Ethernet error control block is a 66B control block (sync header = 10) with a block type of 0x1E and the other eight 7-bit characters equal to 0x1E.

NOTE 2 – The remaining $m-p$ 100G FlexE signals – out of an $m \times 100$ G FlexE group signal – are carried via other ODUflex signals carried in other ODUc_n/OTUC_n/OTSiA signals.

NOTE 3 – For the case that an $m \times 100$ G FlexE group signal is transported via two or more FlexEp- n services, it is possible that one service instance contains a single 100G FlexE signal. This signal could be transported as a FlexE unaware service, but in that case its transfer delay and behaviour under fault conditions would be different than that of the other FlexE aware services. Therefore, this single 100G FlexE signal could be transported as a FlexE1-20 signal; i.e., as a FlexE aware service in which all twenty calendar slots are forwarded (and none are crunched).

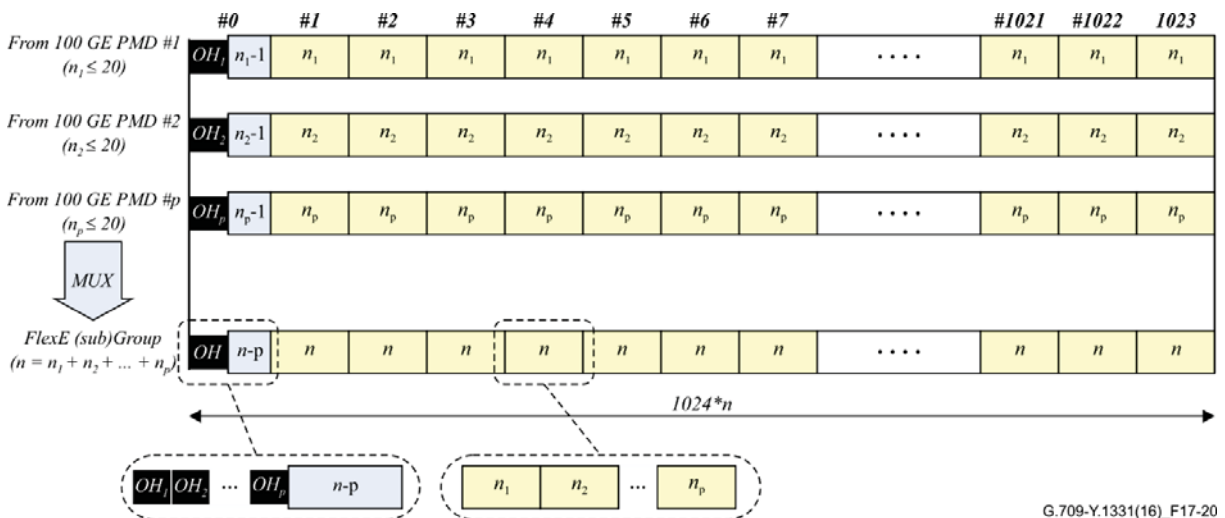


Figure 17-20 – Full/partial rate FlexE signal interleaving

The 66b block stream of the FlexE partial rate (sub)group signal shall be scrambled after rate adaptation and before mapping into the OPUflex. In the reverse operation, following termination of the OPUflex signal, the 66 block stream will be descrambled before being passed to the FlexE deinterleaving, depadding and decrunching process.

A self-synchronizing scrambler with generator polynomial $1 + x^{39} + x^{58}$ shall be used, that is identical to the scrambler specified in clause 49.2.6 [IEEE 802.3]. 66b block stream scrambling is required to provide security against false 66b block delineation (as the two sync header bits bypass the scrambler), the 66b codewords replicating the OTU and ODU frame alignment signal and the 66b

codewords combined with the OTU scrambler pattern used for the interface replicating the OTU and ODU frame alignment signal.

The FlexE partial rate (sub)group signal (with up to ± 100 ppm bit-rate tolerance) is mapped into the OPUflex as per the encoding and stuffing distribution described below. The OPUflex payload bit rate is $100\text{GE_bit_rate} \times 240/239 \times n/20$ kbit/s ± 100 ppm, with $n = n_1 + n_2 + \dots + n_p$. The ODUflex bit rate is $100\text{GE_bit_rate} \times 240/238 \times n/20$ kbit/s ± 100 ppm.

On the transmit side, the stuffing is inserted as 16-byte blocks and systematically controlled using a 15 bit sigma-delta as:

$$100\text{GE_rate} \times (16k-1)/16k \times n \times 1024/(1+20 \times 1023) = 100\text{GE_rate} \times n/20 \times 240/238 \times 238/239 \times (4 \times 238 - 3 - s)/(4 \times 238) \rightarrow s = 3373/23384 \text{ (Numerator=3373, Denominator=23384)}$$

This encoding uses a bitsynchronous generic mapping procedure (BGMP) for the distribution and encoding of the stuffing. The GMP is operated in a special mode, where the C_m value does not come from a synchronizer but is generated deterministically, and $\sum C_{nD}$ is unused ($C_n = C_m$).

As shown in the Figure 17-22, a value of $n = m = 128$ (16-Byte) is chosen. Given the deterministic generation of the stuffing, only two $C_{128}(t)$ values are used for this mapping ($C_{128} = 948$ or 949). Generating $C_{128} = 949$ (vs. $C_{128} = 948$) is deterministic at the source ($s = 3373/23384$, 15-bit sigma-delta).

$C_m(t)$ is deterministically generated using a sigma-delta $C_m=948/C_m=949$ justification distribution with $s = 3373/23384$:

The distribution of the deterministic justification ($C_m(t) = 948$, instead of $C_m(t) = 949$) follows the sigma-delta equation below:

$$- \quad C_m(t) = 948 \quad \text{if } (j \times 3373) \bmod 23384 < 3373 \quad (17-13)$$

$$- \quad C_m(t) = 949 \quad \text{if } (j \times 3373) \bmod 23384 \geq 3373 \quad (17-14)$$

The index 'j' in equations (17-13) and (17-14) enumerates frames in the 23384 frame sequence. It counts from 1 to 23384.

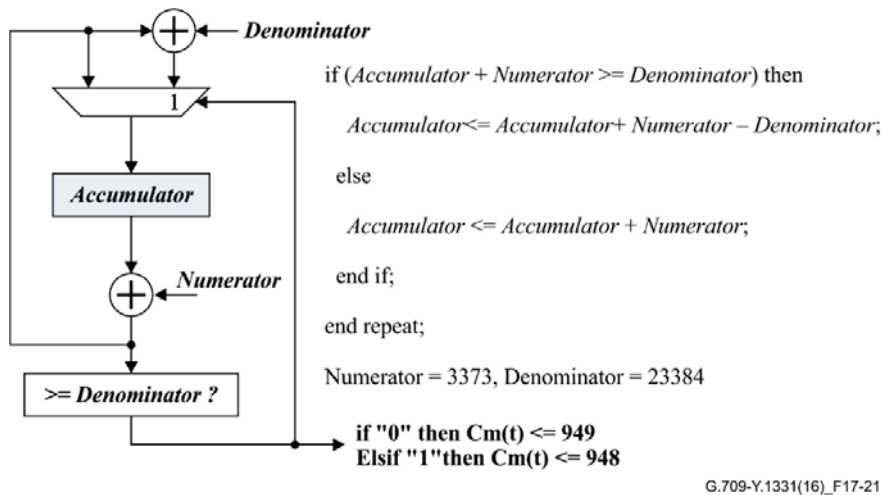


Figure 17-21 – Sigma-delta accumulator for deterministic justification generation

The OPUflex overhead for this mapping consists of a:

- payload structure identifier (PSI) including the payload type (PT) as specified in Table 15-9, the client signal fail (CSF) and 253-p bytes plus 7 bits reserved for future international standardization (RES);

- PSI[3] carries the value of p
- PSI[4] to PSI[3+p] carries the values of n_1 to n_p
- three justification control (JC1, JC2, JC3) bytes carrying the value of GMP overhead C_m ;
- four bytes reserved for future international standardization (RES).

The JC1, JC2 and JC3 bytes consist of a 14-bit C_m field (bits C1, C2, ..., C14), a 1-bit increment indicator (II) field, a 1-bit decrement indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The OPUflex payload for this mapping consists of 4×3808 bytes (Figure 17-22). Blocks of 16 bytes in the OPUflex payload area are numbered from 1 to 952. The OPUflex payload byte numbering for GMP 16-byte (128-bit) blocks is illustrated in Figure 17-22. In row 1 of the OPUflex frame the first 16-byte block will be labelled 1, the next 16-byte block will be labelled 2, etc.

Groups of one hundred twenty-eight successive bits of the client signal are mapped into a 16-byte block of the OPUflex payload area under control of the BGMP data/stuff control mechanism. For the case of $C_{128} = 948$, 16 byte blocks #1, #239, #477 and #715 are carrying stuff bits and the other 16-byte blocks are carrying client bits. For the case of $C_{128} = 949$, 16 byte blocks #1, #318 and #635 are carrying stuff bits and the other 16-byte blocks are carrying client bits. The stuff bits are set to zero.

Scrambled 64b/66b blocks of the client signal are mapped into 66-bits of the OPUflex payload area skipping the 16-byte stuff blocks under control of the BGMP control mechanism. The 66b blocks must begin at OPUflex payload octet bit positions 1, 3, 5, or 7, keeping the 2-bit alignment of each 66b block through the mapping process.

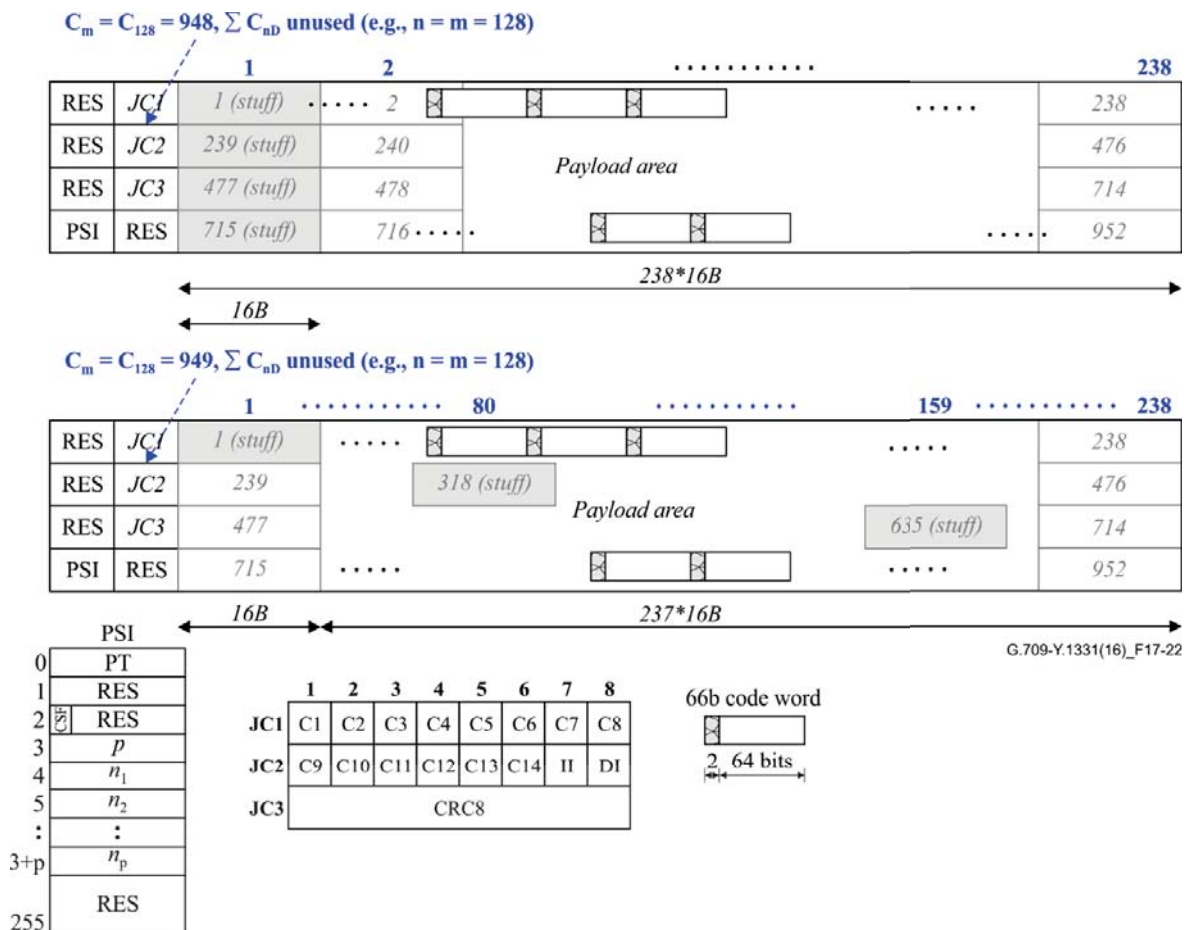


Figure 17-22 – Mapping of the combined FlexE stream into ODUflex via deterministic stuffing and using GMP encoding

During a signal fail condition of an incoming FlexE signal (e.g., in the case of a loss of input signal or FlexE local fault condition), this failed incoming FlexE signal and the p-1 other FlexE signals are not mapped into the payload of the OPUflex. Instead the OPUflex payload is filled with the stuff blocks, 66b blocks that carry a local fault sequence ordered set (as specified in [FlexE IA]) and the OPU CSF bit is set to "1". The OPUflex clock is derived from a local clock. The OPUflex payload signal bit rate shall be within the limits specified in Table 7-3 and neither an OPUflex frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming client signal shall be done without introducing an OPUflex frequency or frame phase discontinuity.

During a signal fail condition of the incoming ODUflex/OPUflex signal (e.g., in the case of an ODUflex-AIS, ODUflex-LCK, ODUflex-OCI condition) or a CSF condition is present in the OPUflex overhead, a stream of 66b blocks, with each block carrying one local fault sequence ordered sets (as specified in [FlexE IA]) is generated as a replacement signal for each of the p lost FlexE signals.

NOTE 4 – A FlexE signal local fault sequence ordered set is a 66B control block (sync header = 10) with a block type of 0x4B, an "O" code of 0x00, a value of 0x01 to indicate a "local fault" in lane 3, and all of the other octets (before scrambling) equal to 0x00.

18 Blank clause

This clause is intentionally left blank.

19 Mapping ODUj signals into the ODTU signal and the ODTU into the OP Uk tributary slots

This clause specifies the multiplexing of:

- ODU0 into OPU1, ODU1 into OPU2, ODU1 and ODU2 into OPU3 using client/server specific asynchronous mapping procedures (AMP);
- other ODUj into OP Uk using a client agnostic generic mapping procedure (GMP).

This ODUj into OP Uk multiplexing is performed in two steps:

- 1) asynchronous mapping of ODUj into optical data tributary unit (ODTU) using either AMP or GMP;
- 2) byte-synchronous mapping of ODTU into one or more OP Uk tributary slots.

19.1 OP Uk tributary slot definition

The OP Uk is divided into a number of tributary slots (TS) and these tributary slots are interleaved within the OP Uk. A tributary slot includes a part of the OP Uk OH area and a part of the OP Uk payload area. The bytes of the ODUj frame are mapped into the ODTU payload area and the ODTU bytes are mapped into the OP Uk tributary slot or slots. The bytes of the ODTU justification overhead are mapped into the OP Uk OH area.

There are two types of tributary slots:

- 1) Tributary slot with a bandwidth of approximately 2.5 Gbit/s; an OP Uk is divided into n tributary slots, numbered 1 to n.
- 2) Tributary slot with a bandwidth of approximately 1.25 Gbit/s; an OP Uk is divided into 2n tributary slots, numbered 1 to 2n.

OPU2 and OPU3 interface ports supporting 1.25 Gbit/s tributary slots must also support the 2.5 Gbit/s tributary slot mode for interworking with interface ports supporting only the 2.5G tributary slot mode (i.e., interface ports compliant with issues of this Recommendation: prior to the definition of 1.25G

tributary slots). When operated in 2.5G tributary slot mode, 1.25G tributary slots "i" and "i+n" (i = 1 to n, n = 4 (OPU2) and n = 16 (OPU3)) function as one 2.5G tributary slot.

19.1.1 OPU2 tributary slot allocation

Figure 19-1 presents the OPU2 2.5G tributary slot allocation and the OPU2 1.25G tributary slot allocation. An OPU2 is divided into four 2.5G tributary slots numbered 1 to 4, or in eight 1.25G tributary slots numbered 1 to 8.

- An OPU2 2.5G tributary slot occupies 25% of the OPU2 payload area. It is a structure with 952 columns by 16 (4×4) rows (see Figures 19-1 and 19-7) plus a tributary slot overhead (TSOH). The four OPU2 TSs are byte interleaved in the OPU2 payload area and the four OPU2 TSOHs are frame interleaved in the OPU2 overhead area.
- An OPU2 1.25G tributary slot occupies 12.5% of the OPU2 payload area. It is a structure with 476 columns by 32 (8×4) rows (see Figures 19-1 and 19-7) plus a tributary slot overhead (TSOH). The eight OPU2 TSs are byte interleaved in the OPU2 payload area and the eight OPU2 TSOHs are frame interleaved in the OPU2 overhead area.

An OPU2 2.5G tributary slot "i" (i = 1,2,3,4) is provided by two OPU2 1.25G tributary slots "i" and "i+4" as illustrated in Figure 19-1.

The tributary slot overhead (TSOH) of OPU2 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU2 frame.

The TSOH for a 2.5G tributary slot is available once every 4 frames. A 4-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 7 and 8 of the MFAS byte as shown in Table 19-1 and Figure 19-1.

The TSOH for a 1.25G tributary slot is available once every 8 frames. An 8-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 6, 7 and 8 of the MFAS byte as shown in Table 19-1 and Figure 19-1.

MFAS bits (6)78	Multi-frame row	Frame row	Column	1	15	16	17	18	19	20	21	22	23	24	25	26	3823	3824
(0)00	1	1																		
	2	2																		
	3	3																		
	4	4																		
(0)01	5	1																		
	6	2																		
	7	3																		
	8	4																		
(0)11	13	1																		
	14	2																		
	15	3																		
	16	4																		
(1)00	17	1																		
	18	2																		
	19	3																		
	20	4																		
(1)11	29	13																		
	30	14																		
	31	15																		
	32	16																		

G.709-Y.1331(12)_F19-1

Figure 19-1 – OPU2 tributary slot allocation

Table 19-1 – OPU2 tributary slot OH allocation

MFAS bits 7 8	TSOH 2.5G TS
0 0	1
0 1	2
1 0	3
1 1	4

MFAS bits 6 7 8	TSOH 1.25G TS
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6
1 1 0	7
1 1 1	8

19.1.2 OPU3 tributary slot allocation

Figure 19-2 presents the OPU3 2.5G tributary slot allocation and the OPU3 1.25G tributary slot allocation. An OPU3 is divided into sixteen 2.5G tributary slots numbered 1 to 16, or in thirty-two 1.25G tributary slots numbered 1 to 32.

- An OPU3 2.5G tributary slot occupies 6.25% of the OPU3 payload area. It is a structure with 238 columns by 64 (16×4) rows (see Figures 19-2 and 19-8) plus a tributary slot overhead (TSOH). The sixteen OPU3 2.5G TSs are byte interleaved in the OPU3 payload area and the sixteen OPU3 TSOHs are frame interleaved in the OPU3 overhead area.
- An OPU3 1.25G tributary slot occupies 3.125% of the OPU3 payload area. It is a structure with 119 columns by 128 (32×4) rows (see Figures 19-2 and 19-8) plus a tributary slot overhead (TSOH). The thirty-two OPU3 1.25G TSs are byte interleaved in the OPU3 payload area and the thirty-two OPU3 TSOHs are frame interleaved in the OPU3 overhead area.

An OPU3 2.5G tributary slot "i" ($i = 1, 2, \dots, 16$) is provided by two OPU3 1.25G tributary slots "i" and "i+16" as illustrated in Figure 19-2.

The tributary slot overhead (TSOH) of OPU3 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU3 frame.

The TSOH for a 2.5G tributary slot is available once every 16 frames. A 16-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 5, 6, 7 and 8 of the MFAS byte as shown in Table 19-2 and Figure 19-2.

The TSOH for a 1.25G tributary slot is available once every 32 frames. A 32-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 4, 5, 6, 7 and 8 of the MFAS byte as shown in Table 19-2 and Figure 19-2.

MFAS bits (4)5678	Multi- frame row	Frame row	Column																																
			1	15	16	17	18	31	32	33	34	47	48	49	50	3823	3824														
(0)0000	1	1																																	
	2	2																																	
	3	3																																	
	4	4																																	
(0)0001	5	1																																	
	6	2																																	
	7	3																																	
	8	4																																	
⋮	⋮																																		
(0)1111	61	1																																	
	62	2																																	
	63	3																																	
	64	4																																	
(1)0000	65	1																																	
	66	2																																	
	67	3																																	
	68	4																																	
⋮	⋮																																		
(1)1111	125	61																																	
	126	62																																	
	127	63																																	
	128	64																																	

G.709-Y.1331(12) F19-2

G.709-Y.1331(12)_F19-2

Figure 19-2 – OPU3 tributary slot allocation

Table 19-2 – OPU3 tributary slot OH allocation

MFAS bits 5 6 7 8	TSOH 2.5G TS	MFAS bits 4 5 6 7 8	TSOH 1.25G TS	MFAS bits 4 5 6 7 8	TSOH 1.25G TS
0 0 0 0	1	0 0 0 0 0	1	1 0 0 0 0	17
0 0 0 1	2	0 0 0 0 1	2	1 0 0 0 1	18
0 0 1 0	3	0 0 0 1 0	3	1 0 0 1 0	19
0 0 1 1	4	0 0 0 1 1	4	1 0 0 1 1	20
0 1 0 0	5	0 0 1 0 0	5	1 0 1 0 0	21
0 1 0 1	6	0 0 1 0 1	6	1 0 1 0 1	22
0 1 1 0	7	0 0 1 1 0	7	1 0 1 1 0	23
0 1 1 1	8	0 0 1 1 1	8	1 0 1 1 1	24
1 0 0 0	9	0 1 0 0 0	9	1 1 0 0 0	25
1 0 0 1	10	0 1 0 0 1	10	1 1 0 0 1	26
1 0 1 0	11	0 1 0 1 0	11	1 1 0 1 0	27
1 0 1 1	12	0 1 0 1 1	12	1 1 0 1 1	28
1 1 0 0	13	0 1 1 0 0	13	1 1 1 0 0	29
1 1 0 1	14	0 1 1 0 1	14	1 1 1 0 1	30
1 1 1 0	15	0 1 1 1 0	15	1 1 1 1 0	31
1 1 1 1	16	0 1 1 1 1	16	1 1 1 1 1	32

19.1.3 OPU1 tributary slot allocation

Figure 19-3 presents the OPU1 1.25G tributary slot allocation. An OPU1 is divided into two 1.25G tributary slots numbered 1 to 2.

- An OPU1 1.25G tributary slot occupies 50% of the OPU1 payload area. It is a structure with 1904 columns by 8 (2×4) rows (see Figure 19-3) plus a tributary slot overhead (TSOH). The two OPU1 1.25G TSs are byte interleaved in the OPU1 payload area and the two OPU1 TSOHs are frame interleaved in the OPU1 overhead area.

The tributary slot overhead (TSOH) of OPU1 tributary slots is located in column 16 plus column 15, rows 1, 2 and 3 of the OPU1 frame.

The TSOH for a 1.25G tributary slot is available once every 2 frames. A 2-frame multiframe structure is used for this assignment. This multiframe structure is locked to bit 8 of the MFAS byte as shown in Table 19-3 and Figure 19-3.

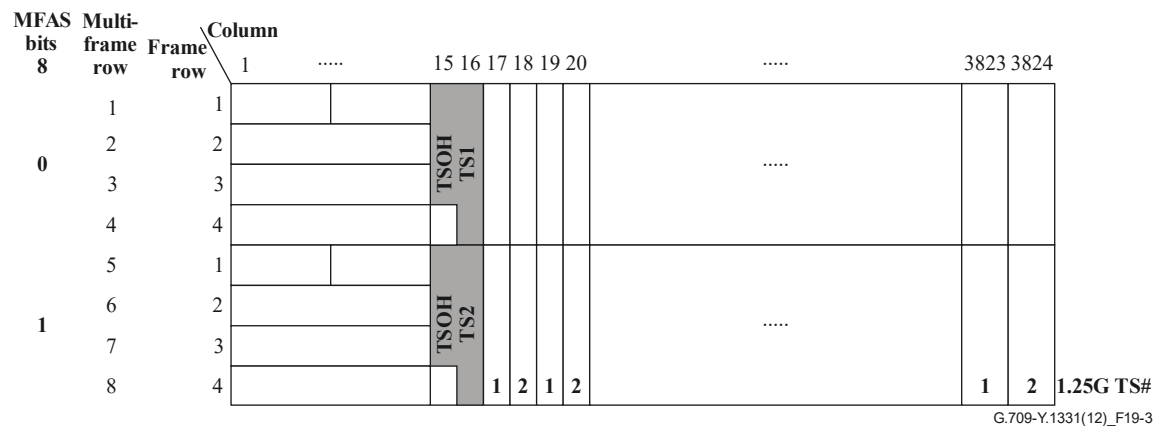
**Figure 19-3 – OPU1 tributary slot allocation**

Table 19-3 – OPU1 tributary slot OH allocation

MFAS bit 8	TSOH 1.25G TS
0	1
1	2

19.1.4 OPU4 tributary slot allocation

Figures 19-4A and 19-4B present the OPU4 1.25G tributary slot allocation. An OPU4 is divided into eighty 1.25G tributary slots (numbered 1 to 80), which are located in columns 17 to 3816, and 8 columns of fixed stuff located in columns 3817 to 3824. The OPU4 frame may be represented in a 320 row by 3810 column format (Figure 19-4A) and in a 160 row by 7620 column format (Figure 19-4B).

- An OPU4 1.25G tributary slot occupies 1.247% of the OPU4 payload area. It is a structure with 95 columns by 160 ($80 \times 4/2$) rows (see Figure 19-4B) plus a tributary slot overhead (TSOH). The eighty OPU4 1.25G TSs are byte interleaved in the OPU4 payload area and the eighty OPU4 TSOHs are frame interleaved in the OPU4 overhead area.

The tributary slot overhead (TSOH) of OPU4 tributary slots is located in rows 1 to 3, columns 15 and 16 of the OPU4 frame.

The TSOH for a 1.25G tributary slot is available once every 80 frames. An 80-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 2, 3, 4, 5, 6, 7 and 8 of the OMFI byte as shown in Table 19-4.

OMFI bits	Multi-frame row	Frame row	Column	1	15	16	17	18	55	56	57	58	95	96	97	98	3815	3816	3817	3818	3819	3820	3821	3822	3823	3824		
0000000	1	1						1	2		39	40	41	42		79	80	1	2	39	40	FS	FS	FS	FS	FS	FS	FS	FS		
	2	2						41	42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	
	3	3						1	2		39	40	41	42		79	80	1	2			39	40	FS	FS	FS	FS	FS	FS	FS	FS	
	4	4						OM FI	41 42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	
0000001	1	1						1	2		39	40	41	42		79	80	1	28	39	40	FS	FS	FS	FS	FS	FS	FS	FS		
	2	2						41	42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	
	3	3						1	2		39	40	41	42		79	80	1	2			39	40	FS	FS	FS	FS	FS	FS	FS	FS	
	4	4						OM FI	41 42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	
⋮																																
1001110	313	1						1	2		39	40	41	42		79	80	1	2	39	40	FS	FS	FS	FS	FS	FS	FS	FS		
	314	2						41	42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	
	315	3						1	2		39	40	41	42		79	80	1	2			39	40	FS	FS	FS	FS	FS	FS	FS	FS	
	316	4						OM FI	41 42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	FS
1001111	317	1						1	2		39	40	41	42		79	80	1	2	39	40	FS	FS	FS	FS	FS	FS	FS	FS		
	318	2						41	42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	FS
	319	3						1	2		39	40	41	42		79	80	1	2			39	40	FS	FS	FS	FS	FS	FS	FS	FS	FS
	320	4						OM FI	41 42		79	80	1	2		39	40	41	42			79	80	FS	FS	FS	FS	FS	FS	FS	FS	FS

G:709-Y:1331(12) F19-4

G.709-Y.1331(12)_F19-4A

Figure 19-4A – OPU4 1.25G tributary slot allocation

OMFI bits 2345678	Multi-frame D-row	Frame row	Column	1	15	16	17	18	55	56	57	58	3815	3816	3817	3818	3819	3820	3821	3822	3823	3824
1	0000000	1	1 + 2			TS1		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
2		3 + 4				PSI		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
3		1 + 2				TS2		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
4	0000001	3 + 4				PSI		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
⋮																									
157	1001110	1 + 2				TS79		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
158		3 + 4				PSI		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
159		1 + 2				TS80		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS
160	1001111	3 + 4				PSI		41 42			79 80	1	2			79 80	FS	FS	FS	FS	FS	FS	FS	FS	FS

G.709-Y.1331(12)_F19-4B

Figure 19-4B – OPU4 tributary slots in 160 row x 7620 column format

Table 19-4 – OPU4 tributary slot OH allocation

OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS	OMFI bits 2 3 4 5 6 7 8	TSOH 1.25G TS
0 0 0 0 0 0 0	1	0 0 1 0 1 0 0	21	0 1 0 1 0 0 0	41	0 1 1 1 1 0 0	61
0 0 0 0 0 0 1	2	0 0 1 0 1 0 1	22	0 1 0 1 0 0 1	42	0 1 1 1 1 0 1	62
0 0 0 0 0 1 0	3	0 0 1 0 1 1 0	23	0 1 0 1 0 1 0	43	0 1 1 1 1 1 0	63
0 0 0 0 0 1 1	4	0 0 1 0 1 1 1	24	0 1 0 1 0 1 1	44	0 1 1 1 1 1 1	64
0 0 0 0 1 0 0	5	0 0 1 1 0 0 0	25	0 1 0 1 1 0 0	45	1 0 0 0 0 0 0	65
0 0 0 0 1 0 1	6	0 0 1 1 0 0 1	26	0 1 0 1 1 0 1	46	1 0 0 0 0 0 1	66
0 0 0 0 1 1 0	7	0 0 1 1 0 1 0	27	0 1 0 1 1 1 0	47	1 0 0 0 0 1 0	67
0 0 0 0 1 1 1	8	0 0 1 1 0 1 1	28	0 1 0 1 1 1 1	48	1 0 0 0 0 1 1	68
0 0 0 1 0 0 0	9	0 0 1 1 1 0 0	29	0 1 1 0 0 0 0	49	1 0 0 0 1 0 0	69
0 0 0 1 0 0 1	10	0 0 1 1 1 0 1	30	0 1 1 0 0 0 1	50	1 0 0 0 1 0 1	70
0 0 0 1 0 1 0	11	0 0 1 1 1 1 0	31	0 1 1 0 0 1 0	51	1 0 0 0 1 1 0	71
0 0 0 1 0 1 1	12	0 0 1 1 1 1 1	32	0 1 1 0 0 1 1	52	1 0 0 0 1 1 1	72
0 0 0 1 1 0 0	13	0 1 0 0 0 0 0	33	0 1 1 0 1 0 0	53	1 0 0 1 0 0 0	73
0 0 0 1 1 0 1	14	0 1 0 0 0 0 1	34	0 1 1 0 1 0 1	54	1 0 0 1 0 0 1	74
0 0 0 1 1 1 0	15	0 1 0 0 0 1 0	35	0 1 1 0 1 1 0	55	1 0 0 1 0 1 0	75
0 0 0 1 1 1 1	16	0 1 0 0 0 1 1	36	0 1 1 0 1 1 1	56	1 0 0 1 0 1 1	76
0 0 1 0 0 0 0	17	0 1 0 0 1 0 0	37	0 1 1 1 0 0 0	57	1 0 0 1 1 0 0	77
0 0 1 0 0 0 1	18	0 1 0 0 1 0 1	38	0 1 1 1 0 0 1	58	1 0 0 1 1 0 1	78
0 0 1 0 0 1 0	19	0 1 0 0 1 1 0	39	0 1 1 1 0 1 0	59	1 0 0 1 1 1 0	79
0 0 1 0 0 1 1	20	0 1 0 0 1 1 1	40	0 1 1 1 0 1 1	60	1 0 0 1 1 1 1	80

19.2 ODTU definition

The optical data tributary unit (ODTU) carries a justified ODU signal. There are two types of ODTUs:

- 1) ODTU_{jk} ((j,k) = {(0,1), (1,2), (1,3), (2,3)}; ODTU01, ODTU12, ODTU13 and ODTU23) in which an ODU_j signal is mapped via the asynchronous mapping procedure (AMP) as defined in clause 19.5;
- 2) ODTU_{k.ts} ((k,ts) = (2,1..8), (3,1..32), (4,1..80)) in which a ODU_j (ODU0, ODU1, ODU2, ODU2e, ODU3, ODUflex) signal is mapped via the generic mapping procedure (GMP) defined in clause 19.6.

Optical data tributary unit jk

The optical data tributary unit jk (ODTU_{jk}) is a structure which consists of an ODTU_{jk} payload area and an ODTU_{jk} overhead area (Figure 19-5). The ODTU_{jk} payload area has c columns and r rows (see Table 19-5) and the ODTU_{jk} overhead area has "ts" times 4 bytes, of which "ts" times 1 byte can carry payload. The ODTU_{jk} is carried in "ts" 1.25G or 2.5G tributary slots of an OPUk.

The location of the ODTU_{jk} overhead depends on the OPU_k tributary slot(s) used when multiplexing the ODTU_{jk} in the OPU_k (see clauses 19.1.1, 19.1.2, 19.1.3). The ts instances of the ODTU_{jk} overhead might not be equally distributed.

The ODTU_{jk} overhead carries the AMP justification overhead as specified in clause 19.4.

NOTE – The 1.25G and 2.5G tributary slot versions of an ODTU12 are identical when the two 1.25G tributary slots carrying the ODTU12 are TS_a and TS_a+4. The 1.25G and 2.5G tributary slot versions of an ODTU13 are identical when the two 1.25G tributary slots carrying the ODTU12 are TS_a and TS_a+16. The 1.25G and 2.5G tributary slot versions of an ODTU23 are identical when the eight 1.25G tributary slots carrying the ODTU23 are TS_a, TS_b, TS_c, TS_d, TS_a+16, TS_b+16, TS_c+16 and TS_d+16.

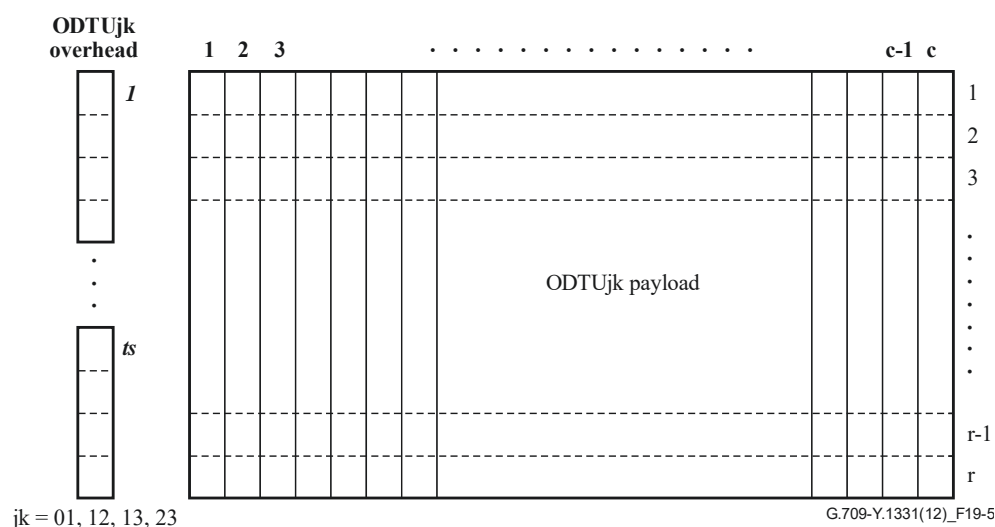


Figure 19-5 – ODTU_{jk} frame formats

Table 19-5 – ODTU_{jk} characteristics for 2.5G and 1.25G tributary slots

2.5G TS	c	r	ts	ODTU _{jk} payload bytes	ODTU _{jk} overhead bytes
ODTU12	952	16	1	15232	1 x 4
ODTU13	238	64	1	15232	1 x 4
ODTU23	952	64	4	60928	4 x 4

1.25G TS	c	r	ts	ODTU _{jk} payload bytes	ODTU _{jk} overhead bytes
ODTU01	1904	8	1	15232	1 x 4
ODTU12	952	32	2	30464	2 x 4
ODTU13	238	128	2	30464	2 x 4
ODTU23	952	128	8	121856	8 x 4

Optical data tributary unit k.ts

The optical data tributary unit k.ts (ODTUK.ts) is a structure which consists of an ODTUK.ts payload area and an ODTUK.ts overhead area (Figure 19-6). The ODTUK.ts payload area has j x ts columns and r rows (see Table 19-6) and the ODTUK.ts overhead area has one times 6 bytes. The ODTUK.ts is carried in "ts" 1.25G tributary slots of an OPU_k.

The location of the ODTUk.ts overhead depends on the OPUk tributary slot used when multiplexing the ODTUk.ts in the OPUk (see clauses 19.1.1, 19.1.2, 19.1.4). The single instance of an ODTUk.ts overhead is located in the OPUk TSOH of the last OPUk tributary slot allocated to the ODTUk.ts.

The ODTUk.ts overhead carries the GMP justification overhead as specified in clause 19.4.

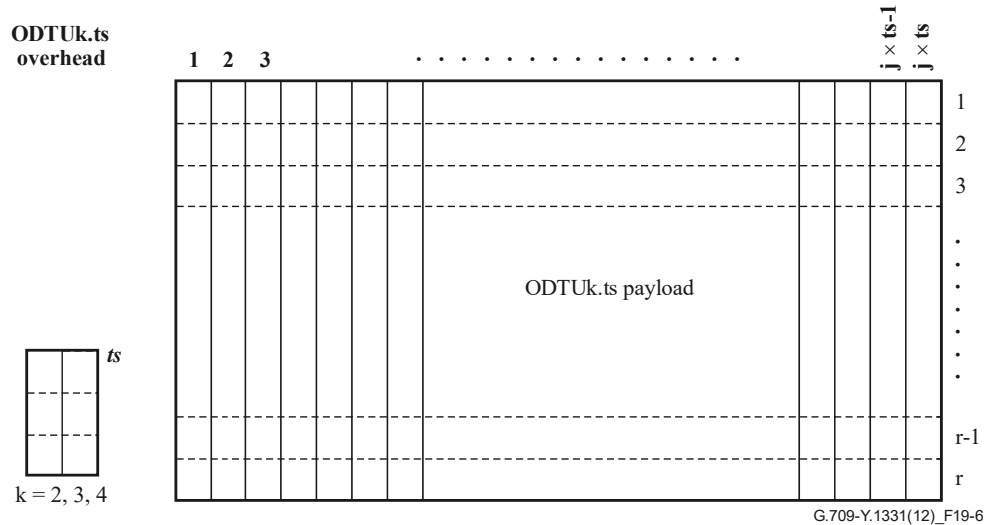


Figure 19-6 – ODTUk.ts frame formats

Table 19-6 – ODTUk.ts characteristics

	j	r	ts	ODTUk.ts payload bytes	ODTUk.ts overhead bytes
ODTU2.ts	476	32	1 to 8	$15232 \times ts$	1×6
ODTU3.ts	119	128	1 to 32	$15232 \times ts$	1×6
ODTU4.ts	95	160	1 to 80	$15200 \times ts$	1×6

19.3 Multiplexing ODTU signals into the OPUk

Multiplexing an ODTU01 signal into an OPU1 is realized by mapping the ODTU01 signal in one of the two OPU1 1.25G tributary slots.

Multiplexing an ODTU12 signal into an OPU2 is realized by mapping the ODTU12 signal in one of the four OPU2 2.5G tributary slots or in two (of the eight) arbitrary OPU2 1.25G tributary slots: OPU2 TSa and TSb with $1 \leq a < b \leq 8$.

Multiplexing an ODTU13 signal into an OPU3 is realized by mapping the ODTU13 signal in one of the sixteen OPU3 2.5G tributary slots or in two (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa and TSb with $1 \leq a < b \leq 32$.

Multiplexing an ODTU23 signal into an OPU3 is realized by mapping the ODTU23 signal in four (of the sixteen) arbitrary OPU3 2.5G tributary slots: OPU3 TSa, TSb, TSc and TSd with $1 \leq a < b < c < d \leq 16$ or in eight (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa, TSb, TSc, TSd, TSe, TSf, TSg and TSh with $1 \leq a < b < c < d < e < f < g < h \leq 32$.

NOTE – a, b, c, d, e, f, g and h do not have to be sequential ($a = i$, $b = i+1$, $c = i+2$, $d = i+3$, $e = i+4$, $f = i+5$, $g = i+6$, $h = i+7$); the values can be arbitrarily selected to prevent bandwidth fragmentation.

Multiplexing an ODTU2.ts signal into an OPU2 is realized by mapping the ODTU2.ts signal in ts (of the eight) arbitrary OPU2 1.25G tributary slots: OPU2 TSa, TSb, .., TS_p with $1 \leq a < b < \dots < p \leq 8$.

Multiplexing an ODTU3.ts signal into an OPU3 is realized by mapping the ODTU3.ts signal in ts (of the thirty-two) arbitrary OPU3 1.25G tributary slots: OPU3 TSa, TSb, .. , TSq with $1 \leq a < b < \dots < q \leq 32$.

Multiplexing an ODTU4.ts signal into an OPU4 is realized by mapping the ODTU4.ts signal in ts (of the eighty) arbitrary OPU4 1.25G tributary slots: OPU4 TSa, TSb, .. , TSr with $1 \leq a < b < \dots < r \leq 80$.

The OPU_k overhead for these multiplexed signals consists of a payload type (PT), the multiplex structure identifier (MSI), the OPU_k multiframe identifier (k=4), the OPU_k tributary slot overhead carrying the ODTU overhead and depending on the ODTU type one or more bytes reserved for future international standardization.

19.3.1 ODTU12 mapping into one OPU2 tributary slot

A byte of the ODTU12 payload signal is mapped into a byte of an OPU2 2.5G TS #i (i = 1,2,3,4) payload area, as indicated in Figure 19-7 (left). A byte of the ODTU12 overhead is mapped into a TSOH byte within column 16 of the OPU2 2.5G TS #i.

A byte of the ODTU12 signal is mapped into a byte of one of two OPU2 1.25G TS #A,B (A,B = 1,2,...,8) payload areas, as indicated in Figure 19-7 (right). A byte of the ODTU12 overhead is mapped into a TSOH byte within column 16 of the OPU2 1.25G TS #a,b.

The remaining OPU2 TSOH bytes in column 15 are reserved for future international standardization.

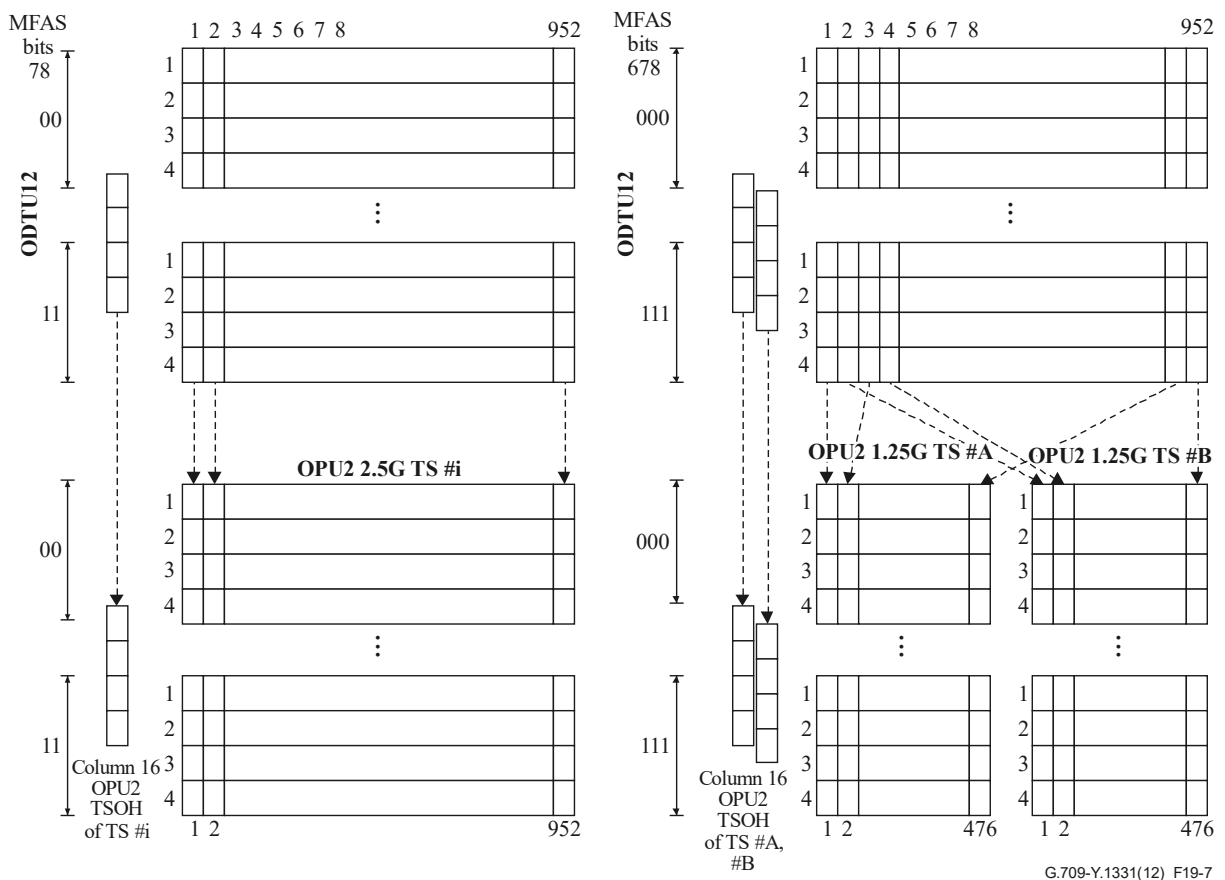


Figure 19-7 – Mapping of ODTU12 into one OPU2 2.5G tributary slot (left) and two OPU2 1.25G tributary slots (right)

19.3.2 ODTU13 mapping into one OPU3 tributary slot

A byte of the ODTU13 signal is mapped into a byte of an OPU3 2.5G TS #i ($i = 1, 2, \dots, 16$) payload area, as indicated in Figure 19-8 (left). A byte of the ODTU13 overhead is mapped into a TSOH byte within column 16 of the OPU3 2.5G TS #i.

A byte of the ODTU13 signal is mapped into a byte of one of two OPU3 1.25G TS #A, B ($A, B = 1, 2, \dots, 32$) payload areas, as indicated in Figure 19-8 (right). A byte of the ODTU13 overhead is mapped into a TSOH byte within column 16 of the OPU3 1.25G TS #a,b.

The remaining OPU3 TSOH bytes in column 15 are reserved for future international standardization.

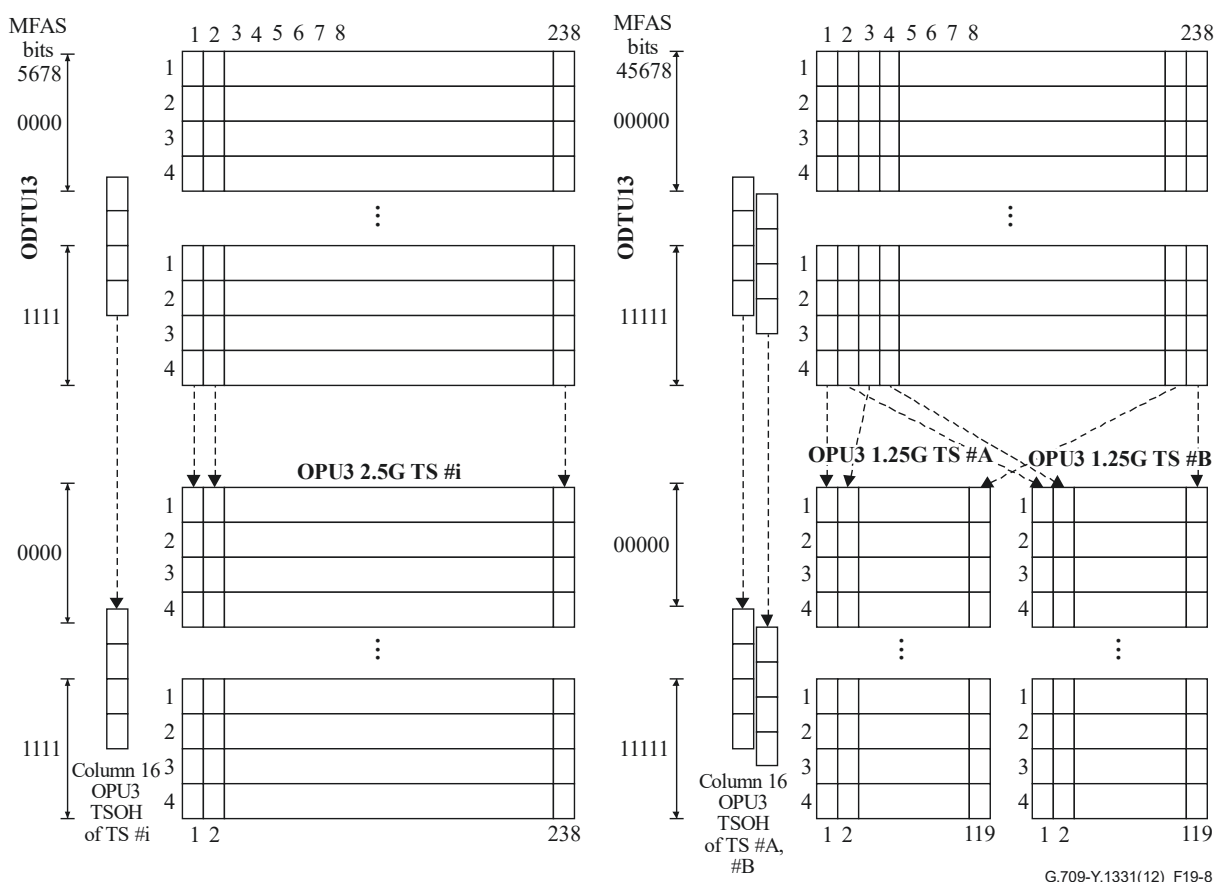


Figure 19-8 – Mapping of ODTU13 into one OPU3 2.5G tributary slot (left) and two OPU3 1.25G tributary slots (right)

19.3.3 ODTU23 mapping into four OPU3 tributary slots

A byte of the ODTU23 signal is mapped into a byte of one of four OPU3 2.5G TS #A,B,C,D ($A, B, C, D = 1, 2, \dots, 16$) payload areas, as indicated in Figure 19-9 (top). A byte of the ODTU23 overhead is mapped into a TSOH byte within column 16 of the OPU3 TS #a,b,c,d.

A byte of the ODTU23 signal is mapped into a byte of one of eight OPU3 1.25G TS #A, B, C, D, E, F, G, H ($A, B, C, D, E, F, G, H = 1, 2, \dots, 32$) payload areas, as indicated in Figure 19-9 (bottom). A byte of the ODTU23 overhead is mapped into a TSOH byte within column 16 of the OPU3 1.25G TS #a,b,c,d,e,f,g,h.

The remaining OPU3 TSOH bytes in column 15 are reserved for future international standardization.

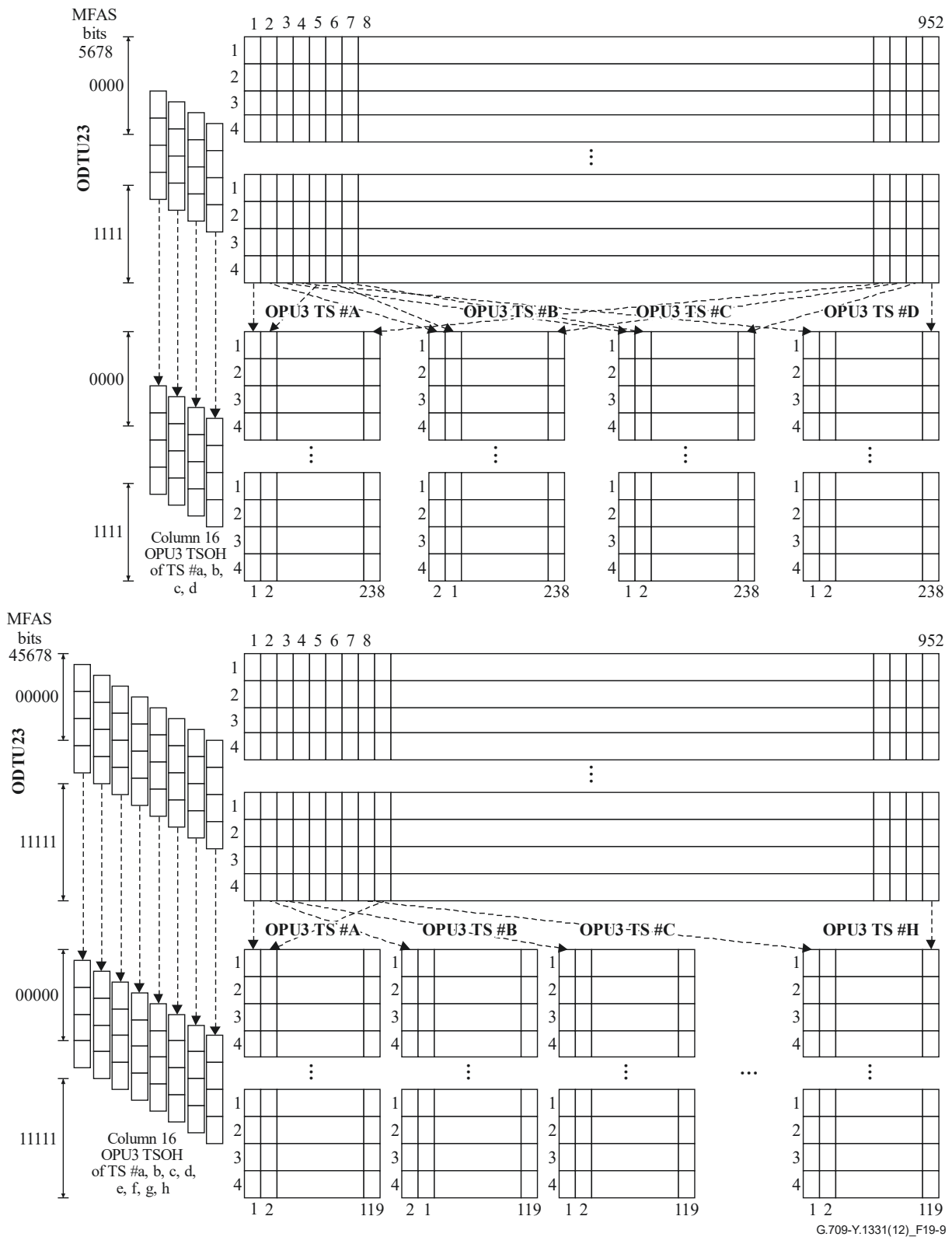


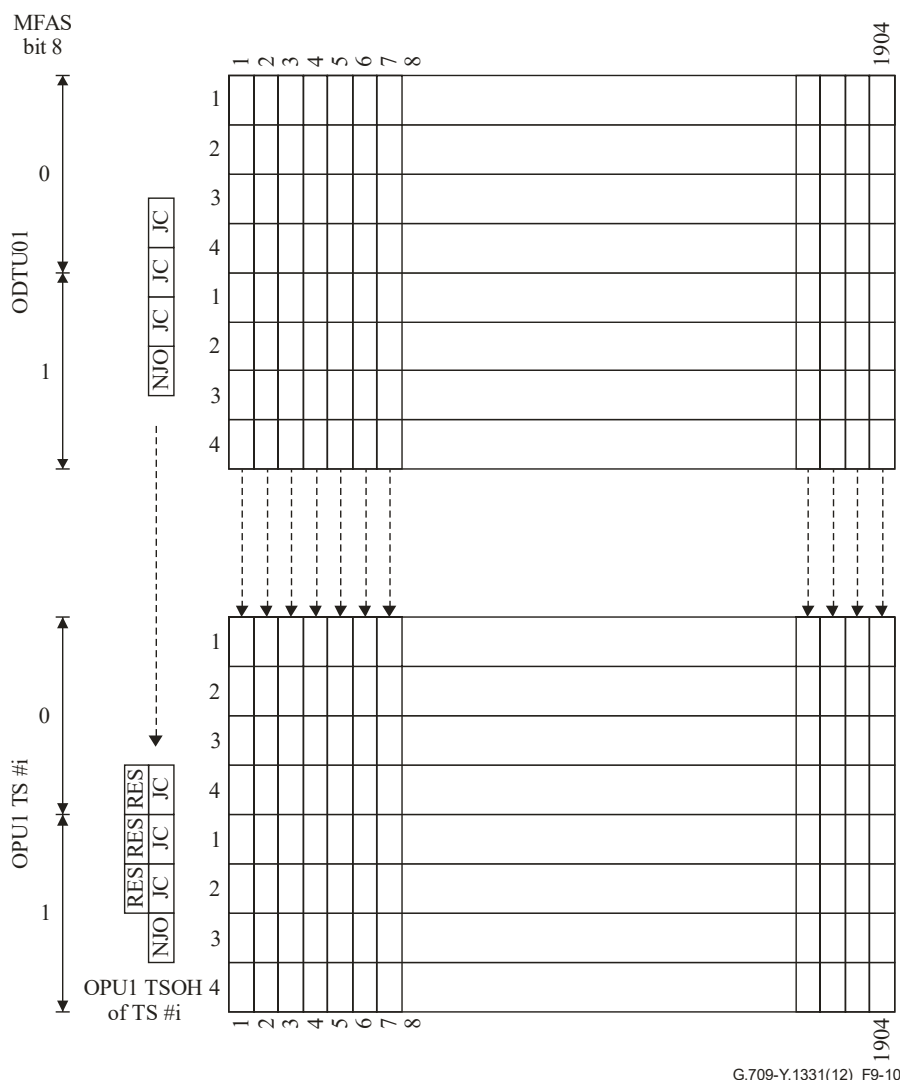
Figure 19-9 – Mapping of ODTU23 into 4 OPU3 2.5G tributary slots (#A, #B, #C, #D with A<B<C<D) (top) and 8 OPU3 1.25G tributary slots (#A, #B, #C, #D, #E, #F, #G, #H with A<B<C<D<E<F<G<H) (bottom)

19.3.4 ODTU01 mapping into one OPU1 1.25G tributary slot

A byte of the ODTU01 signal is mapped into a byte of an OPU1 1.25G TS #i ($i = 1, 2$), as indicated in Figure 19-10 for a group of 4 rows out of the ODTU01.

A byte of the ODTU01 TSOH is mapped into a TSOH byte within column 16 of the OPU1 1.25G TS #i.

The remaining OPU1 TSOH bytes in column 15 are reserved for future international standardization.



G.709-Y.1331(12)_F9-10

Figure 19-10 – Mapping of ODTU01 (excluding JOH) into OPU1 1.25G tributary slot

19.3.5 ODTU2.ts mapping into ts OPU2 1.25G tributary slots

A byte of the ODTU2.ts payload signal is mapped into a byte of an OPU2 1.25G TS #i ($i = 1, \dots, ts$) payload area, as indicated in Figure 19-11.

A byte of the ODTU2.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU2 1.25G tributary slot allocated to the ODTU2.ts.

The remaining OPU2 TSOH bytes are reserved for future international standardization.

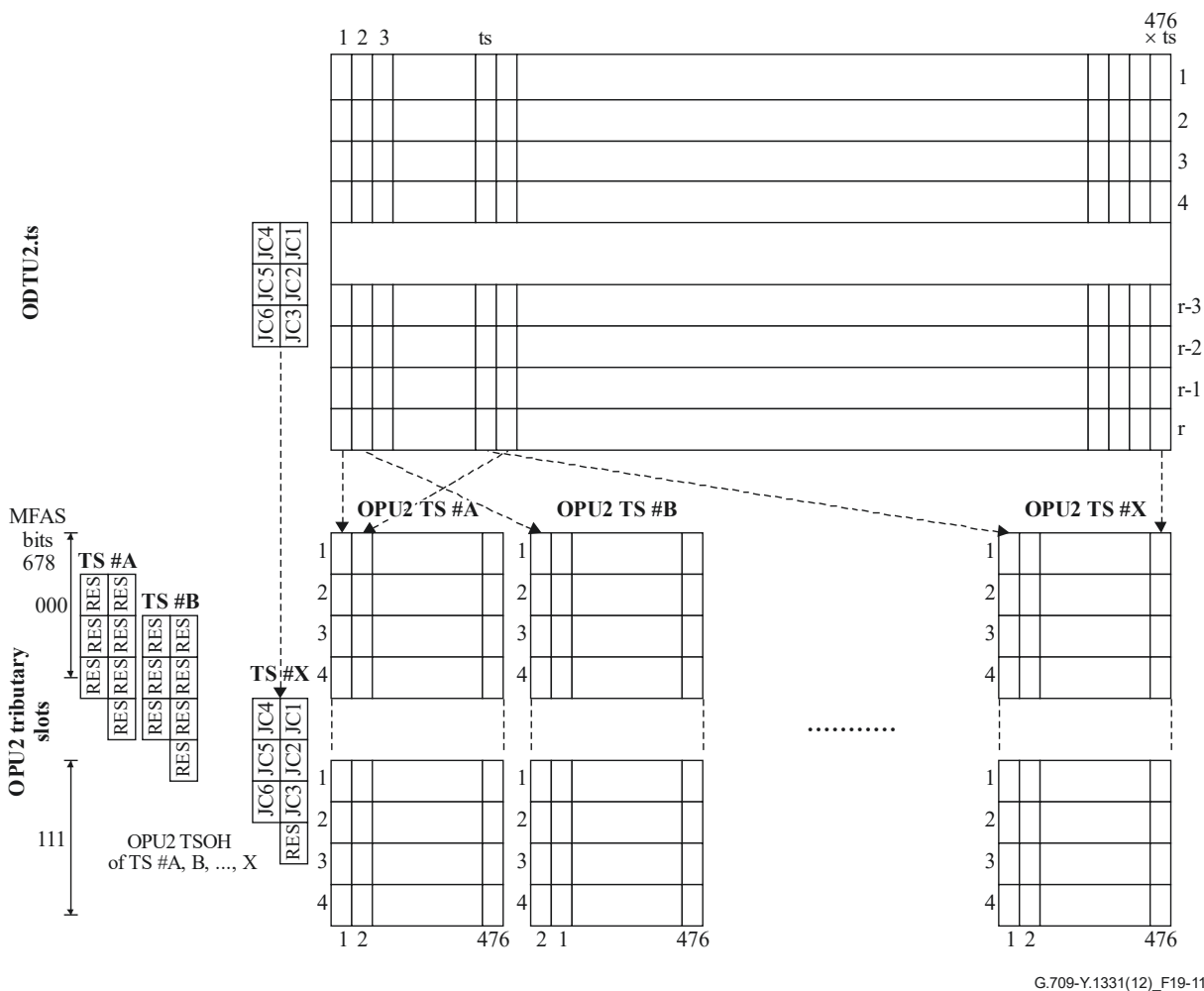


Figure 19-11 – Mapping of ODTU2.ts into 'ts' OPU2 1.25G tributary slots

19.3.6 ODTU3.ts mapping into ts OPU3 1.25G tributary slots

A byte of the ODTU3.ts payload signal is mapped into a byte of an OPU3 1.25G TS #i (i = 1,...,ts) payload area, as indicated in Figure 19-12.

A byte of the ODTU3.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU3 1.25G tributary slot allocated to the ODTU3.ts.

The remaining OPU3 TSOH bytes are reserved for future international standardization.

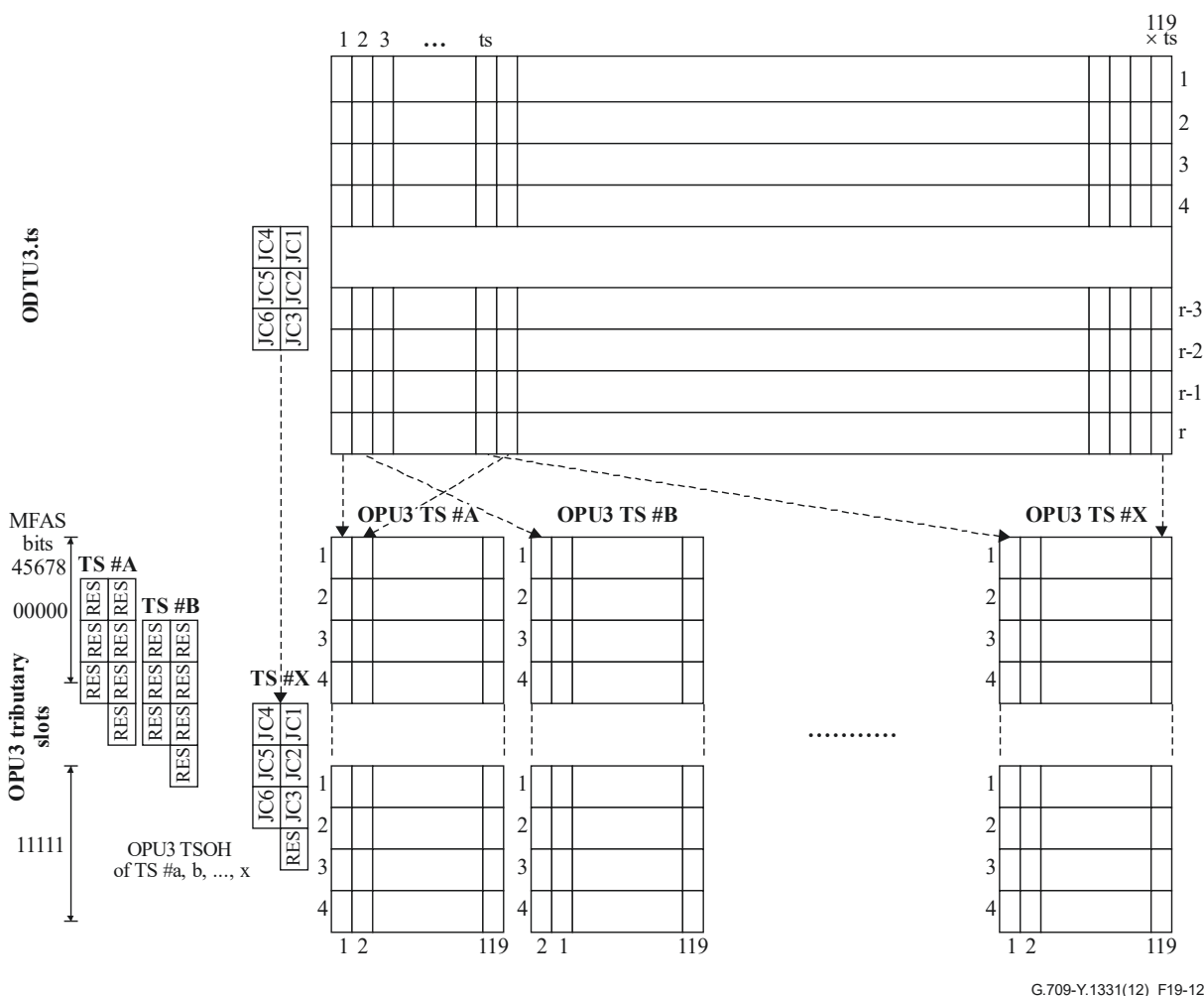


Figure 19-12 – Mapping of ODTU3.ts into 'ts' OPU3 1.25G tributary slots

19.3.7 ODTU4.ts mapping into ts OPU4 1.25G tributary slots

A byte of the ODTU4.ts payload signal is mapped into a byte of an OPU4 1.25G TS #*i* (*i* = 1,...,ts) payload area, as indicated in Figure 19-13.

A byte of the ODTU4.ts overhead is mapped into a TSOH byte within columns 15 and 16, rows 1 to 3 of the last OPU4 1.25G tributary slot allocated to the ODTU4.ts.

The remaining OPU4 TSOH bytes are reserved for future international standardization.

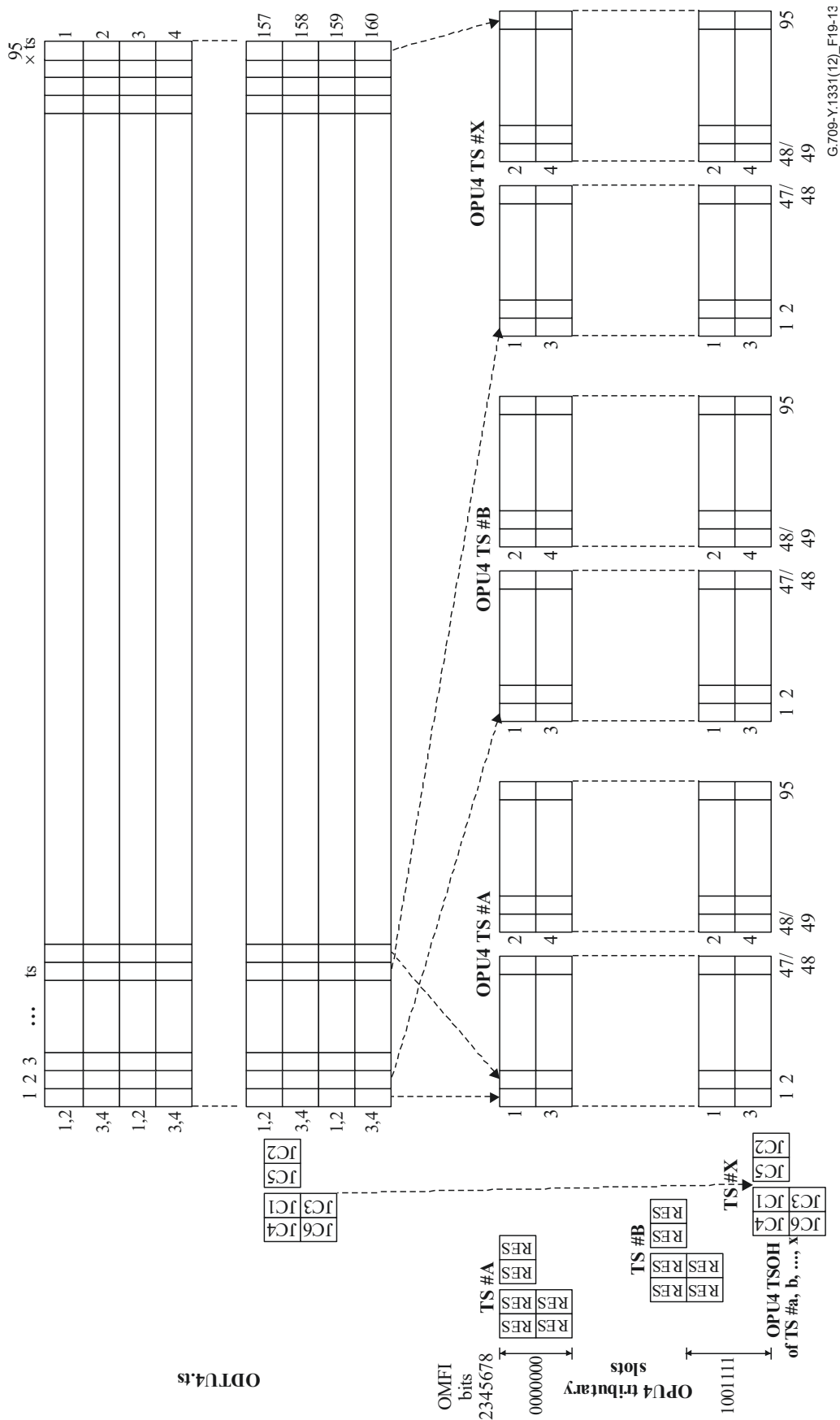


Figure 19-13 – Mapping of ODTU4.ts into 'ts' OPU4 1.25G tributary slots

19.4 OPUk multiplex overhead and ODTU justification overhead

The OPUk ($k=1,2,3,4$) multiplex overhead consists of a multiplex structure identifier (MSI) and an ODTU overhead. The OPUk ($k=4$) multiplex overhead contains an OPU multiframe identifier (OMFI).

The OPUk MSI overhead locations are shown in Figures 19-14A, 19-14B and 19-14C and the OMFI overhead location is shown in Figure 19-14C.

ODTUjk overhead

The ODTUjk overhead carries the AMP justification overhead consisting of justification control (JC) and negative justification opportunity (NJO) signals in column 16 of rows 1 to 4. ODTUjk overhead bytes in column 15 rows 1, 2 and 3 are reserved for future international standardization.

The ODTUjk overhead consists of 3 bytes of justification control (JC) and 1 byte of negative justification opportunity (NJO) overhead. The JC and NJO overhead locations are shown in Figures 19-14A and 19-14B. In addition, two or n times two positive justification overhead bytes (PJO1, PJO2) are located in the ODTUjk payload area. Note that the PJO1 and PJO2 locations are multiframe, ODUj and OPUk tributary slot dependent.

The PJO1 for an ODU1 in OPU2 or OPU3 2.5G tributary slot #i ($i: 1..4$ or $1..16$ respectively) is located in the first column of OPUk 2.5G tributary slot #i (OPUk column $16+i$) and the PJO2 is located in the second column of OPUk 2.5G tributary slot #i (OPU2 column $20+i$, OPU3 column $32+i$) in frame #i of the four or sixteen frame multiframe.

EXAMPLE – ODU1 in OPU2 or OPU3 TS(1): PJO1 in column $16+1=17$, PJO2 in column $20+1=21$ (OPU2) and $32+1=33$ (OPU3). ODU1 in OPU2 TS(4): PJO1 in column $16+4=20$, PJO2 in column $20+4=24$. ODU1 in OPU3 TS(16): PJO1 in column $16+16=32$, PJO2 in column $32+16=48$.

The four PJO1s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #a (OPU3 column $16+a$) in frames #a, #b, #c and #d of the sixteen frame multiframe. The four PJO2s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #b (OPU3 column $16+b$) in frames #a, #b, #c and #d of the sixteen frame multiframe. Figure 19-14A presents an example with four ODU2s in the OPU3 mapped into 2.5G tributary slots (1,5,9,10), (2,3,11,12), (4,14,15,16) and (6,7,8,13).

EXAMPLE – ODU2 in OPU3 TS(1,2,3,4): PJO1 in column $16+1=17$, PJO2 in column $16+2=18$. ODU2 in OPU3 TS(13,14,15,16): PJO1 in column $16+13=29$, PJO2 in column $16+14=30$.

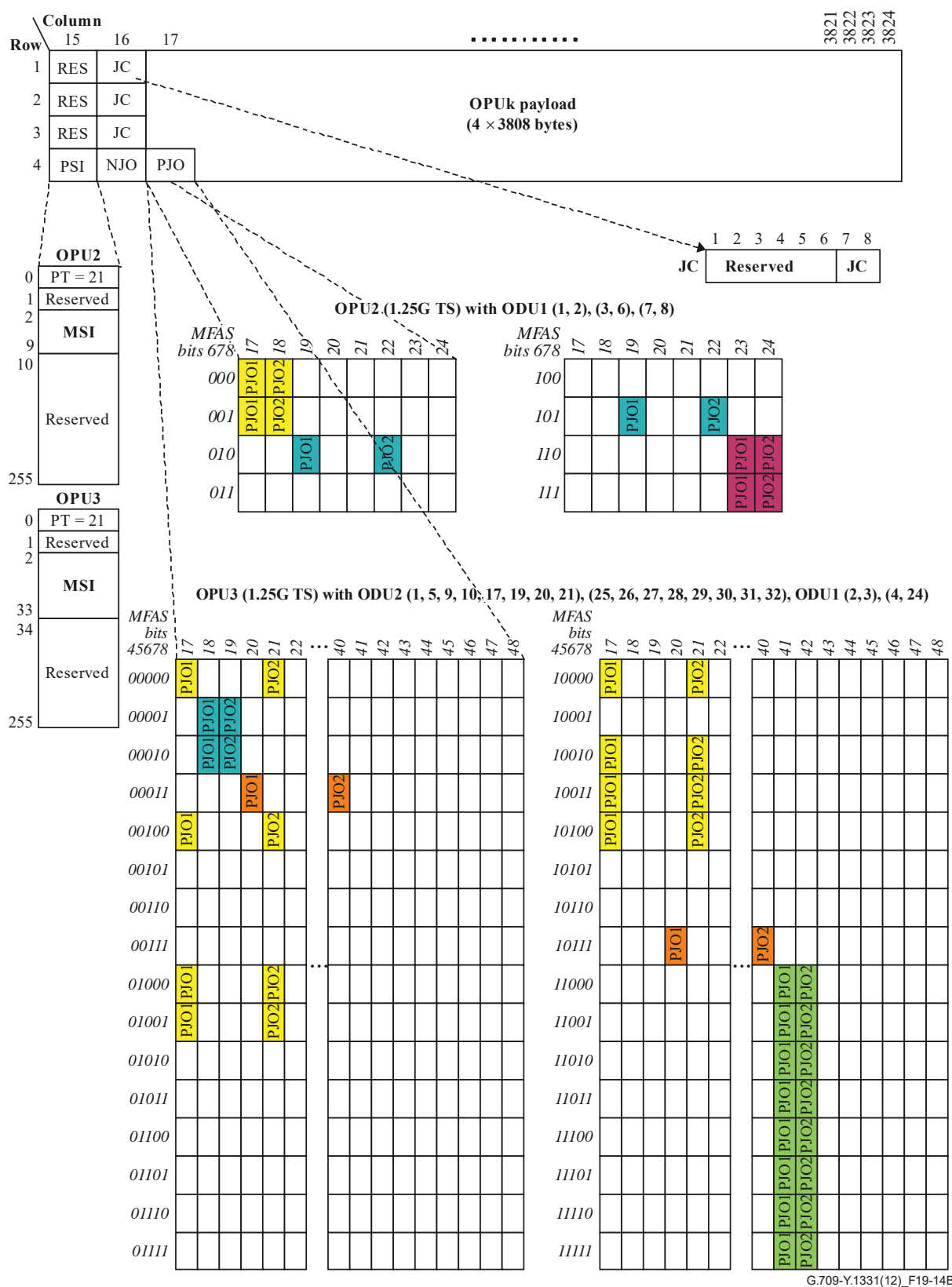
The PJO1 for an ODU0 in OPU1 1.25G tributary slot #i ($i: 1,2$) is located in the first column of OPU1 1.25G tributary slot #i (OPU1 column $16+i$) and the PJO2 is located in the second column of OPU1 1.25G tributary slot #i (OPU1 column $18+i$) in frame #i of the two frame multiframe.

The PJO1 for an ODU1 in OPU2 or OPU3 1.25G tributary slots #a and #b ($a: 1..7$ or $1..31$ respectively; $b: 2..8$ or $2..32$ respectively) is located in the first column of OPUk 1.25G tributary slot #a (OPUk column $16+a$) and the PJO2 is located in the first column of OPUk 1.25G tributary slot #b (OPUk column $16+b$) in frames #a and #b of the eight or thirty-two frame multiframe.

EXAMPLE – ODU1 in OPU2 or OPU3 TS(1,2): PJO1 in column $16+1=17$, PJO2 in column $16+2=18$. ODU1 in OPU2 TS(7,8): PJO1 in column $16+7=23$, PJO2 in column $16+8=24$. ODU1 in OPU3 TS(31,32): PJO1 in column $16+31=47$, PJO2 in column $16+32=48$.

The eight PJO1s for an ODU2 in OPU3 1.25G tributary slots #a, #b, #c, #d, #e, #f, #g and #h are located in the first column of OPU3 1.25G tributary slot #a (OPU3 column $16+a$) in frames #a, #b, #c, #d, #e, #f, #g and #h of the thirty-two frame multiframe. The eight PJO2s for an ODU2 in OPU3 1.25G tributary slots #a, #b, #c, #d, #e, #f, #g and #h are located in the first column of OPU3 1.25G tributary slot #b (OPU3 column $16+b$) in frames #a, #b, #c, #d, #e, #f, #g and #h of the thirty-two frame multiframe. Figure 19-14B presents an example with two ODU2s and two ODU1s in the OPU3 mapped into 1.25G tributary slots (1,5,9,10,17,19,20,21), (25,26,27,28,29,30,31,32), (2,3) and (4,24).

EXAMPLE – ODU2 in OPU3 TS(1,2,3,4,5,6,7,8): PJO1 in column $16+1=17$, PJO2 in column $16+2=18$. ODU2 in OPU3 TS(25,26,27,28,29,30,31,32): PJO1 in column $16+25=41$, PJO2 in column $16+26=42$.



**Figure 19-14B – OPUk (k=2,3) multiplex overhead associated with an ODTUjk only
(payload type = 21)**

ODTUK.ts overhead

The ODTUk.ts overhead carries the GMP justification overhead consisting of 3 bytes of justification control (JC1, JC2, JC3) which carry the 14-bit GMP C_m information and client/ODU specific 3 bytes of justification control (JC4, JC5, JC6) which carry the 10-bit GMP ΣC_{8D} information.

The JC1, JC2, JC3, JC4, JC5 and JC6 overhead locations are shown in Figure 19-14C.

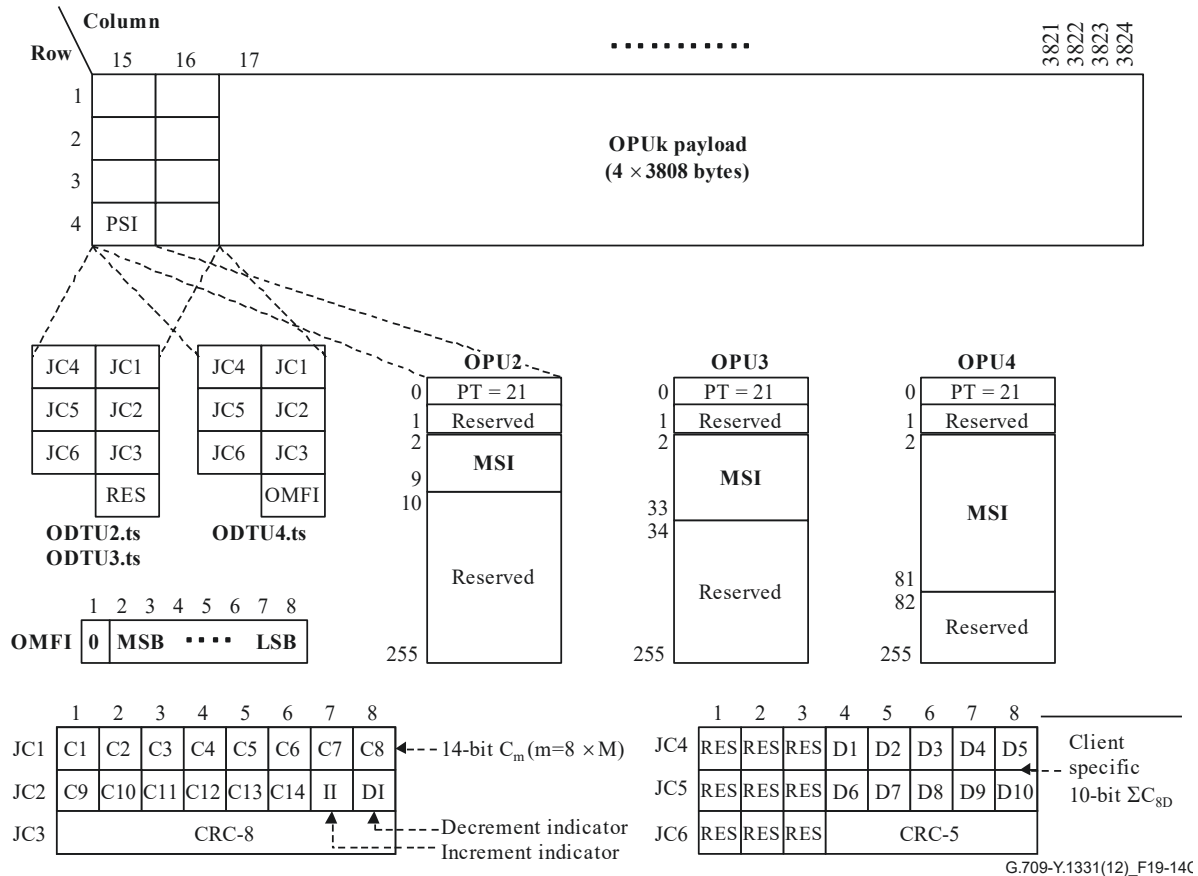


Figure 19-14C – OPUk (k=2,3,4) multiplex overhead associated with an ODTUk.ts (payload type = 21)

19.4.1 OPUk multiplex structure identifier (MSI)

The OPUk (k=1,2,3,4) multiplex structure identifier (MSI) overhead, which encodes the ODU multiplex structure in the OPU, is located in the mapping specific area of the PSI signal (refer to Figure 19-14A for the MSI location in OPUk with PT=20, refer to Figures 19-14B and 19-14C for the MSI location in OPUk with PT=21). The MSI has an OPU and tributary slot (2.5G, 1.25G) specific length (OPU1: 2 bytes, OPU2: 4 or 8 bytes, OPU3: 16 or 32 bytes, OPU4: 80 bytes) and indicates the ODTU content of each tributary slot (TS) of an OPU. One byte is used for each TS.

19.4.1.1 OPU2 multiplex structure identifier (MSI) – Payload type 20

For the 4 OPU2 2.5G tributary slots four bytes of the PSI are used (PSI[2] .. PSI[5]) as MSI bytes as shown in Figures 19-14A and 19-15. The MSI indicates the ODTU content of each tributary slot of the OPU2. One byte is used for each tributary slot.

- The ODTU type in bits 1 and 2 is fixed to 00 to indicate the presence of an ODTU12.
- The tributary port # indicates the port number of the ODU1 that is being transported in this 2.5G TS; the assignment of ports to tributary slots is fixed, the port number equals the tributary slot number.

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	00				00	0000			<i>TS1</i>
<i>PSI[3]</i>	00				00	0001			<i>TS2</i>
<i>PSI[4]</i>	00				00	0010			<i>TS3</i>
<i>PSI[5]</i>	00				00	0011			<i>TS4</i>

Figure 19-15 – OPU2-MSI coding – Payload type 20

19.4.1.2 OPU3 multiplex structure identifier (MSI) – Payload type 20

For the 16 OPU3 2.5G tributary slots 16 bytes of the PSI are used (*PSI[2]* .. *PSI[17]*) as MSI bytes as shown in Figures 19-14A, 19-16A and 19-16B. The MSI indicates the ODTU content of each tributary slot of the OPU3. One byte is used for each tributary slot.

- The ODTU type in bits 1 and 2 indicates if the OPU3 TS is carrying ODTU13 or ODTU23. The default ODTU type is ODTU13; it is present when either a tributary slot carries an ODTU13, or is not allocated to carry an ODTU. Refer to Appendix V for some examples.
- The tributary port # in bits 3 to 8 indicates the port number of the ODTU13/23 that is being transported in this 2.5G TS; for the case of ODTU23, a flexible assignment of tributary ports to tributary slots is possible, for the case of ODTU13, this assignment is fixed, the tributary port number equals the tributary slot number. ODTU23 tributary ports are numbered 1 to 4.

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	ODTU type		Tributary Port #						<i>TS1</i>
<i>PSI[3]</i>	ODTU type		Tributary Port #						<i>TS2</i>
<i>PSI[4]</i>	ODTU type		Tributary Port #						<i>TS3</i>
<i>PSI[5]</i>	ODTU type		Tributary Port #						<i>TS4</i>
<i>PSI[6]</i>	ODTU type		Tributary Port #						<i>TS5</i>
<i>PSI[7]</i>	ODTU type		Tributary Port #						<i>TS6</i>
<i>PSI[8]</i>	ODTU type		Tributary Port #						<i>TS7</i>
<i>PSI[9]</i>	ODTU type		Tributary Port #						<i>TS8</i>
<i>PSI[10]</i>	ODTU type		Tributary Port #						<i>TS9</i>
<i>PSI[11]</i>	ODTU type		Tributary Port #						<i>TS10</i>
<i>PSI[12]</i>	ODTU type		Tributary Port #						<i>TS11</i>
<i>PSI[13]</i>	ODTU type		Tributary Port #						<i>TS12</i>
<i>PSI[14]</i>	ODTU type		Tributary Port #						<i>TS13</i>
<i>PSI[15]</i>	ODTU type		Tributary Port #						<i>TS14</i>
<i>PSI[16]</i>	ODTU type		Tributary Port #						<i>TS15</i>
<i>PSI[17]</i>	ODTU type		Tributary Port #						<i>TS16</i>

Figure 19-16A – OPU3-MSI coding – Payload type 20

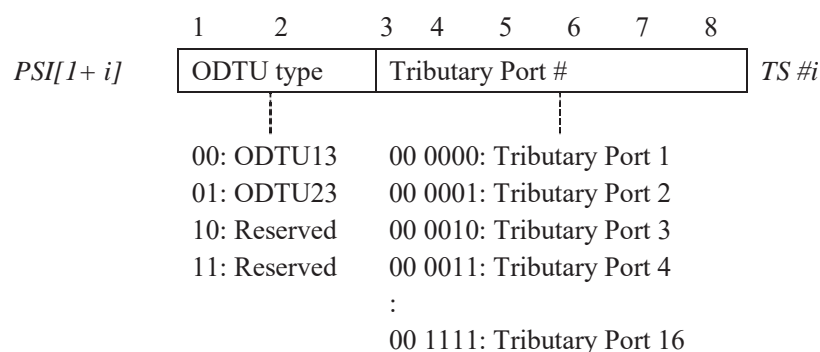


Figure 19-16B – OPU3 MSI coding – Payload type 20

19.4.1.3 OPU1 multiplex structure identifier (MSI) – Payload type 20

For the 2 OPU1 1.25G tributary slots 2 bytes of the PSI are used ($PSI[2]$, $PSI[3]$) as MSI bytes as shown in Figures 19-14A and 19-17. The MSI indicates the ODTU content of each tributary slot of the OPU1. One byte is used for each tributary slot.

- The ODTU type in bits 1 and 2 is fixed to 11 to indicate the presence of an ODTU01.
- The tributary port # in bits 3 to 8 indicates the port number of the ODTU01 that is being transported in this 1.25G TS; the assignment of ports to tributary slots is fixed, the port number equals the tributary slot number.

	1	2	3	4	5	6	7	8	1.25G TS
$PSI[2]$	11		00 0000						$TS1$
$PSI[3]$	11		00 0001						$TS2$

Figure 19-17 – OPU1 MSI coding – Payload type 20

19.4.1.4 OPU4 multiplex structure identifier (MSI) – Payload type 21

For the eighty OPU4 1.25G tributary slots 80 bytes of the PSI are used ($PSI[2]$ to $PSI[81]$) as MSI bytes as shown in Figures 19-14C, 19-18A and 19-18B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

- The TS occupation bit 1 indicates if the tributary slot is allocated or unallocated.
- The tributary port # in bits 2 to 8 indicates the port number of the ODTU4.ts that is being transported in this TS; for the case of an ODTU4.ts carried in two or more tributary slots, a flexible assignment of tributary port to tributary slots is possible. ODTU4.ts tributary ports are numbered 1 to 80. The value is set to all-0s when the occupation bit has the value 0 (tributary slot is unallocated).

	1	2	3	4	5	6	7	8	1.25G TS
<i>PSI[2]</i>	TS occupied	Tributary Port #							<i>TS1</i>
<i>PSI[3]</i>	TS occupied	Tributary Port #							<i>TS2</i>
<i>PSI[4]</i>	TS occupied	Tributary Port #							<i>TS3</i>
<i>PSI[5]</i>	TS occupied	Tributary Port #							<i>TS4</i>
<i>PSI[6]</i>	TS occupied	Tributary Port #							<i>TS5</i>
<i>PSI[7]</i>	TS occupied	Tributary Port #							<i>TS6</i>
<i>PSI[8]</i>	TS occupied	Tributary Port #							<i>TS7</i>
<i>PSI[9]</i>	TS occupied	Tributary Port #							<i>TS8</i>
:	:	:							:
:	:	:							:
<i>PSI[81]</i>	TS occupied	Tributary Port #							<i>TS80</i>

Figure 19-18A – OPU4 1.25G TS MSI coding – Payload type 21

	1	2	3	4	5	6	7	8	
<i>PSI[1 + i]</i>	Occupation	Tributary Port #							<i>TS #i</i>
0 : Unallocated		000 0000: Tributary Port 1							
1 : Allocated		000 0001: Tributary Port 2							
		000 0010: Tributary Port 3							
		000 0011: Tributary Port 4							
		:							
		100 1111: Tributary Port 80							

Figure 19-18B – OPU4 MSI coding – Payload type 21

19.4.1.5 OPU2 multiplex structure identifier (MSI) – Payload type 21

For the eight OPU2 1.25G tributary slots 8 bytes of the PSI (*PSI[2]* to *PSI[9]*) are used as MSI bytes as show in Figures 19-14B, 19-19A and 19-19B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

- The ODTU type in bits 1 and 2 indicates if the OPU2 1.25G TS is carrying an ODTU12 or ODTU2.ts. The default ODTU type is 11 (unallocated); it is present when a tributary slot is not allocated to carry an ODTU.
- The tributary port # in bits 3 to 8 indicates the port number of the ODTU that is being transported in this TS; a flexible assignment of tributary ports to tributary slots is possible, ODTU12 tributary ports are numbered 1 to 4, and ODTU2.ts tributary ports are numbered 1 to 8. The value is set to all-0s when the ODTU type has the value 11 (tributary slot is unallocated).

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	ODTU type		Tributary Port #						<i>TS1</i>
<i>PSI[3]</i>	ODTU type		Tributary Port #						<i>TS2</i>
<i>PSI[4]</i>	ODTU type		Tributary Port #						<i>TS3</i>
<i>PSI[5]</i>	ODTU type		Tributary Port #						<i>TS4</i>
<i>PSI[6]</i>	ODTU type		Tributary Port #						<i>TS5</i>
<i>PSI[7]</i>	ODTU type		Tributary Port #						<i>TS6</i>
<i>PSI[8]</i>	ODTU type		Tributary Port #						<i>TS7</i>
<i>PSI[9]</i>	ODTU type		Tributary Port #						<i>TS8</i>

Figure 19-19A – OPU2 MSI coding – Payload type 21

	1	2	3	4	5	6	7	8	
<i>PSI[1 + i]</i>	ODTU type		Tributary Port #						<i>TS #i</i>
	00: ODTU12		00 0000: Tributary Port 1						
	01: Reserved		00 0001: Tributary Port 2						
	10: ODTU2.ts		00 0010: Tributary Port 3						
	11: Unallocated		00 0011: Tributary Port 4						
			:						
			00 0111: Tributary Port 8						

Figure 19-19B – OPU2 MSI coding – Payload type 21

19.4.1.6 OPU3 with 1.25G tributary slots (payload type 21) multiplex structure identifier (MSI)

For the thirty-two OPU3 1.25G tributary slots 32 bytes of the PSI (*PSI[2]* to *PSI[33]*) are used as MSI bytes as shown in Figures 19-14B, 19-20A and 19-20B. The MSI indicates the ODTU content of each tributary slot of an OPU. One byte is used for each tributary slot.

- The ODTU type in bits 1 and 2 indicates if the OPU3 1.25G TS is carrying an ODTU13, ODTU23 or ODTU3.ts. The default ODTU type is 11 (unallocated); it is present when a tributary slot is not allocated to carry an ODTU.
- The tributary port # in bits 3 to 8 indicates the port number of the ODTU that is being transported in this TS; a flexible assignment of tributary ports to tributary slots is possible, ODTU13 tributary ports are numbered 1 to 16, ODTU23 tributary ports are numbered 1 to 4 and ODTU3.ts tributary ports are numbered 1 to 32. The value is set to all-0s when the ODTU type has the value 11 (tributary slot is unallocated).

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	ODTU type		Tributary Port #						<i>TS1</i>
<i>PSI[3]</i>	ODTU type		Tributary Port #						<i>TS2</i>
<i>PSI[4]</i>	ODTU type		Tributary Port #						<i>TS3</i>
<i>PSI[5]</i>	ODTU type		Tributary Port #						<i>TS4</i>
<i>PSI[6]</i>	ODTU type		Tributary Port #						<i>TS5</i>
:	:		:						:
:	:		:						:
<i>PSI[33]</i>	ODTU type		Tributary Port #						<i>TS32</i>

Figure 19-20A – OPU3 MSI coding – Payload type 21

	1	2	3	4	5	6	7	8	
<i>PSI[1 + i]</i>	ODTU type		Tributary Port #						<i>TS #i</i>
	00: ODTU13		00 0000: Tributary Port 1						
	01: ODTU23		00 0001: Tributary Port 2						
	10: ODTU3.ts		00 0010: Tributary Port 3						
	11: Unallocated		00 0011: Tributary Port 4						
			:						
			01 1111: Tributary Port 32						

Figure 19-20B – OPU3 MSI coding – Payload type 21**19.4.2 OPUk payload structure identifier reserved overhead (RES)**

253 (OPU1), 251 or 247 (OPU2), 239 or 223 (OPU3) and 175 (OPU4) bytes are reserved in the OPUk PSI for future international standardization. These bytes are located in PSI[1] and PSI[4] (OPU1), PSI[6] or PSI[10] (OPU2), PSI[18] or PSI[34] (OPU3), PSI[82] (OPU4) to PSI[255] of the OPUk overhead. These bytes are set to all-0s.

19.4.3 OPUk multiplex justification overhead (JOH)

Two mapping procedures are used for the mapping of ODU_j: either the asynchronous mapping procedure (AMP) or generic mapping procedure (GMP) into ODTU_{jk} or ODTU_k.ts, respectively. AMP uses ODU_j and OPUk specific fixed stuff and justification opportunity definitions (ODTU_{jk}). GMP uses ODU_j and OPUk independent stuff and justification opportunity definitions (ODTU_k.ts). Stuff locations within an ODTU_k.ts are determined by means of a formula which is specified in clause 19.4.3.2.

19.4.3.1 Asynchronous mapping procedures (AMP)

The justification overhead (JOH) located in column 16 of the OPUk (k=1,2,3) as indicated in Figures 19-14A and 19-14B consists of three justification control (JC) bytes and one negative justification opportunity (NJO) byte. The three JC bytes are located in rows 1, 2 and 3. The NJO byte is located in row 4.

Bits 7 and 8 of each JC byte are used for justification control. The other six bits are reserved for future international standardization.

19.4.3.2 Generic mapping procedure (GMP)

The justification overhead (JOH) for the generic mapping procedure consists of two groups of three bytes of justification control; the general (JC1, JC2, JC3) and the client to ODU mapping specific (JC4, JC5, JC6). Refer to Figure 19-14C.

The JC1, JC2 and JC3 bytes consist of a 14-bit C_m field (bits C1, C2, ..., C14), a 1-bit increment indicator (II) field, a 1-bit decrement indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The JC4, JC5 and JC6 bytes consist of a 10-bit ΣC_{nD} field (bits D1, D2, ..., D10), a 5-bit CRC-5 field which contains an error check code over bits 4 to 8 in the JC4, JC5 and JC6 fields and nine bits reserved for future international standardization (RES).

The value of 'm' in C_m is $8 \times \text{'ts'}$ (number of tributary slots occupied by the ODTUk.ts).

The value of 'n' represents the timing granularity of the GMP C_n parameter, which is also present in ΣC_{nD} . The value of n is 8.

The value of C_m controls the distribution of groups of 'ts' ODUj data bytes into groups of 'ts' ODTUk.ts payload bytes. Refer to clause 19.6 and Annex D for further specification of this process.

The value of ΣC_{nD} provides additional 'n'-bit timing information, which is necessary to control the jitter and wander performance experienced by the ODUj signal.

The value of C_n (i.e., number of client n-bit data entities per OPUCn multiframe) is computed as follows: $C_n(t) = m \times C_m(t) + (\Sigma C_{nD}(t) - \Sigma C_{nD}(t-1))$. Note that the value C_{nD} is effectively an indication of the amount of data in the mapper's virtual queue that it could not send during that multiframe due to it being less than an M-byte word. For the case where the value of ΣC_{nD} in a multiframe 't' is corrupted, it is possible to recover from such error in the next multiframe 't+1'.

19.4.4 OPU multiframe identifier overhead (OMFI)

An OPU4 multiframe identifier (OMFI) byte is defined in row 4, column 16 of the OPU4 overhead (Figure 19-21). The value of bits 2 to 8 of the OMFI byte will be incremented each OPU4 frame to provide an 80 frame multiframe for the multiplexing of ODUj signals into the OPU4.

NOTE – It is an option to align the OMFI = 0 position with MFAS = 0 position every 1280 (the least common multiple of 80 and 256) frame periods.

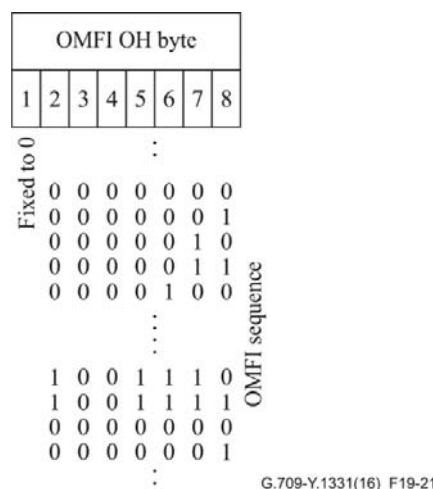


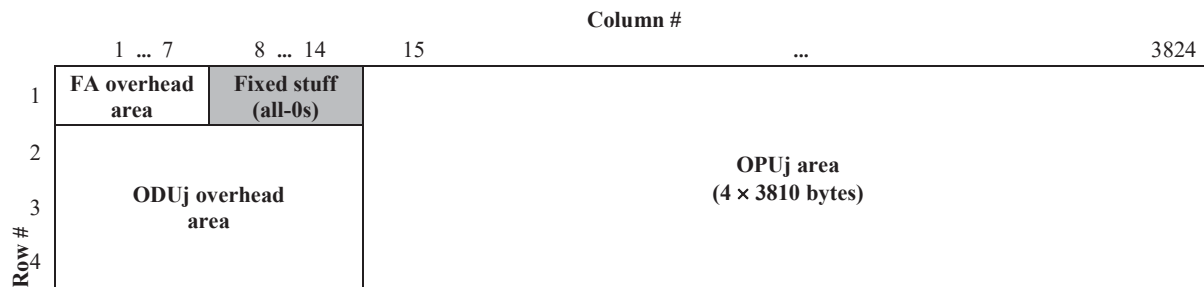
Figure 19-21 – OPU4 multiframe identifier (OMFI) overhead

19.5 Mapping ODUj into ODTUjk

The mapping of ODUj signals (with up to ± 20 ppm bit-rate tolerance) into the ODTUjk signal ((j,k) = {(0,1), (1,2); (1,3), (2,3)}) is performed as an asynchronous mapping.

NOTE 1 – The maximum bit-rate tolerance between OPUk and the ODUj signal clock, which can be accommodated by this mapping scheme is -130 to $+65$ ppm (ODU0 into OPU1), -113 to $+83$ ppm (ODU1 into OPU2), -96 to $+101$ ppm (ODU1 into OPU3) and -95 to $+101$ ppm (ODU2 into OPU3).

The ODUj signal is extended with a frame alignment overhead as specified in clauses 15.6.2.1 and 15.6.2.2 and an all-0s pattern in the OTUj overhead field (see Figure 19-22).



**Figure 19-22 – Extended ODUj frame structure
(FA OH included, OTUj OH area contains fixed stuff)**

The OPUk signal and therefore the ODTUjk (k = 1,2,3) signals are created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the ODUj (j = 0,1,2) client signals.

The extended ODUj (j = 0,1,2) signal is mapped into the ODTUjk (k = 1,2,3) using an asynchronous mapping with $-1/0/+1/+2$ positive/negative/zero (pnz) justification scheme.

An extended ODUj byte is mapped into an ODTUjk byte.

The asynchronous mapping process generates the JC, NJO, PJO1 and PJO2 according to Table 19-7. The de-mapping process interprets JC, NJO, PJO1 and PJO2 according to Table 19-7. Majority vote (two out of three) shall be used to make the justification decision in the de-mapping process to protect against an error in one of the three JC signals.

Table 19-7 – JC, NJO, PJO1 and PJO2 generation and interpretation

JC 7 8	NJO	PJO1	PJO2	Interpretation
0 0	justification byte	data byte	data byte	no justification (0)
0 1	data byte	data byte	data byte	negative justification (−1)
1 0 (Note)	justification byte	justification byte	justification byte	double positive justification (+2)
1 1	justification byte	justification byte	data byte	positive justification (+1)
NOTE – Note that this code is not used for the case of ODU0 into OPU1.				

The value contained in NJO, PJO1 and PJO2 when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming ODU_j client signal (e.g., OTU_j-LOF), this failed incoming signal will contain the ODU_j-AIS signal as specified in clause 16.5.1. This ODU_j-AIS is then mapped into the ODTU_{jk}.

For the case where the ODU_j is received from the output of a fabric (ODU connection function), the incoming signal may contain (in the case of an open matrix connection), the ODU_j-OCI signal as specified in clause 16.5.2. This ODU_j-OCI signal is then mapped into the ODTU_{jk}.

NOTE 2 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such a unit is intentionally absent (i.e., not installed), the associated ODTU_{jk} signals should carry an ODU_j-OCI signal. If such a unit is installed but temporarily removed as part of a repair action, the associated ODTU_{jk} signal should carry an ODU_j-AIS signal.

The de-mapping of ODU_j signals from the ODTU_{jk} signal ($j = 0,1,2$; $k = 1,2,3$) is performed by extracting the extended ODU_j signal from the OPU_k under control of its justification overhead (JC, NJO, PJO1, PJO2).

NOTE 3 – For the case where the ODU_j signal is output as an OTU_j signal, frame alignment of the extracted extended ODU_j signal is to be recovered to allow frame synchronous mapping of the ODU_j into the OTU_j signal.

During a signal fail condition of the incoming ODU_k/OPU_k signal (e.g., in the case of an ODU_k-AIS, ODU_k-LCK, ODU_k-OCI condition) the ODU_j-AIS pattern as specified in clause 16.5.1 is generated as a replacement signal for the lost ODU_j signal.

19.5.1 Mapping ODU1 into ODTU12

A byte of the ODU1 signal is mapped into an information byte of the ODTU12 (see Figure 19-23A). Once per 4 OPU2 frames, it is possible to perform either a positive or a negative justification action. The frame in which justification can be performed is related to the TSOH of the OPU2 2.5G TS in which the ODTU12 is mapped (Figure 19-1). Figure 19-23A shows the case with mapping in OPU2 2.5G TS1.

A byte of the ODU1 signal is mapped into an information byte of the ODTU12 (see Figure 19-23B). Twice per 8 OPU2 frames, it is possible to perform either a positive or a negative justification action. The frames in which justification can be performed are related to the TSOH of the OPU2 1.25G TSs in which the ODTU12 is mapped (Figure 19-1). Figure 19-23B shows the case with mapping in OPU2 1.25G TS2 and TS4.

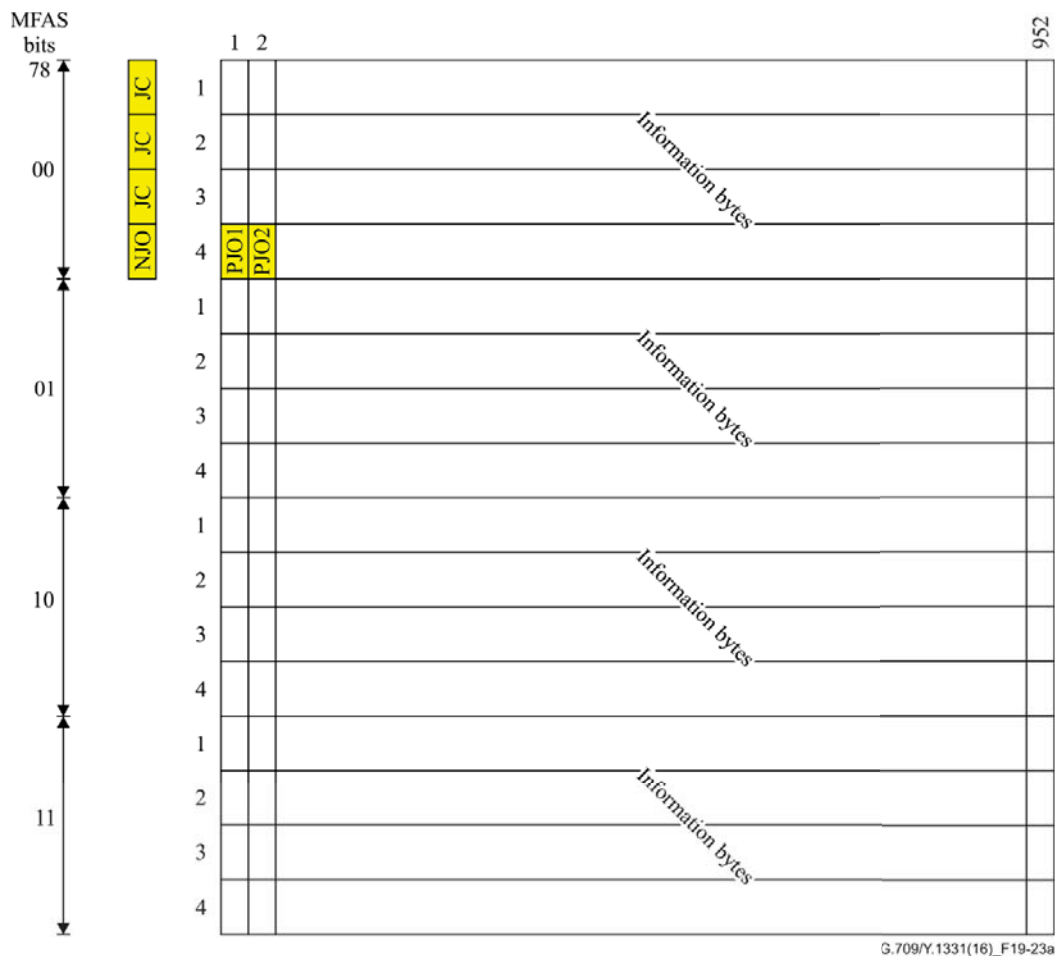
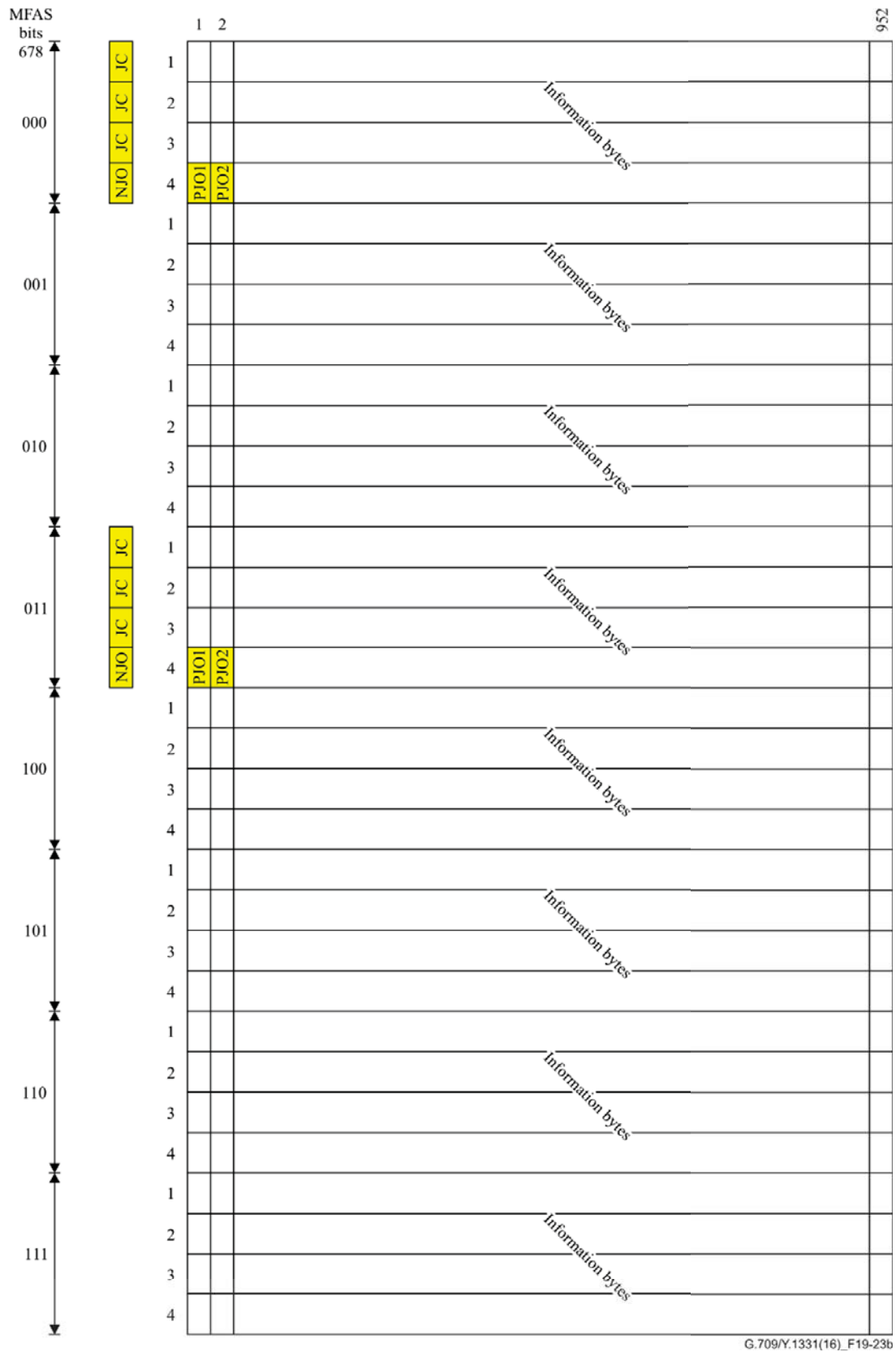
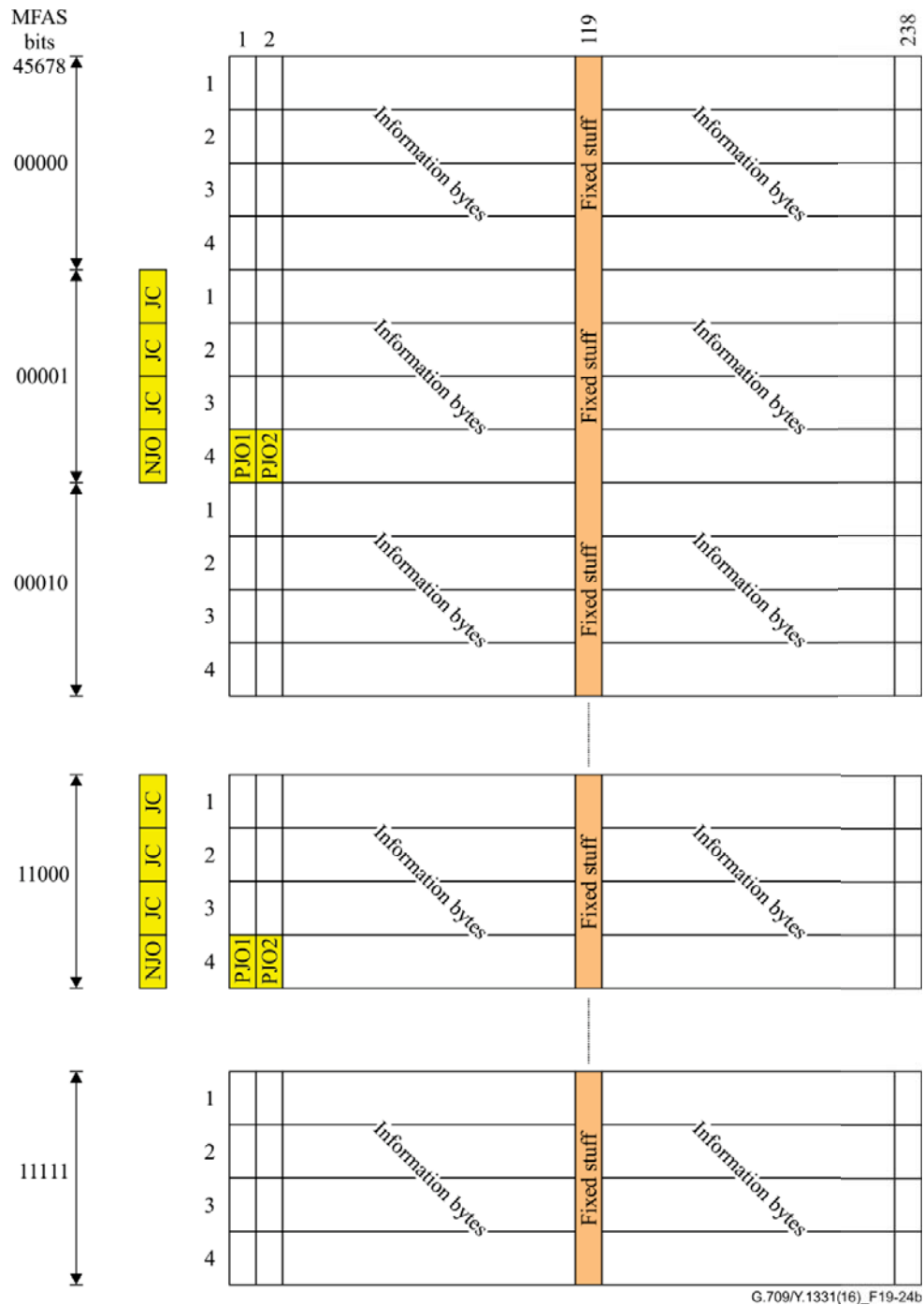


Figure 19-23A – ODTU12 frame format and mapping of ODU1 (mapping in 2.5G TS1)





G.709/Y.1331(16)_F19-24b

**Figure 19-24B – ODTU13 frame format and mapping of ODU1
(mapping in 1.25G TS2 and TS25)**

19.5.3 Mapping ODU2 into ODTU23

A byte of the ODU2 signal is mapped into an information byte of the ODTU23 (Figure 19-25A). Four times per sixteen OPU3 frames, it is possible to perform either a positive or a negative justification action. The four frames in which justification can be performed are related to the TSOH of the OPU3 2.5G TSs in which the ODTU23 is mapped (Figure 19-2). Figure 19-25A shows the case with mapping in OPU3 2.5G TS1, TS5, TS9 and TS10.

A byte of the ODU2 signal is mapped into an information byte of the ODTU23 (see Figure 19-25B). Eight times per 32 OPU3 frames, it is possible to perform either a positive or a negative justification action. The frames in which justification can be performed are related to the TSOH of the OPU3 1.25G TSs in which the ODTU23 is mapped (Figure 19-2). Figure 19-25B shows the case with mapping in OPU3 1.25G TS 1, 2, 5, 9, 10, 25, 26 and 32.

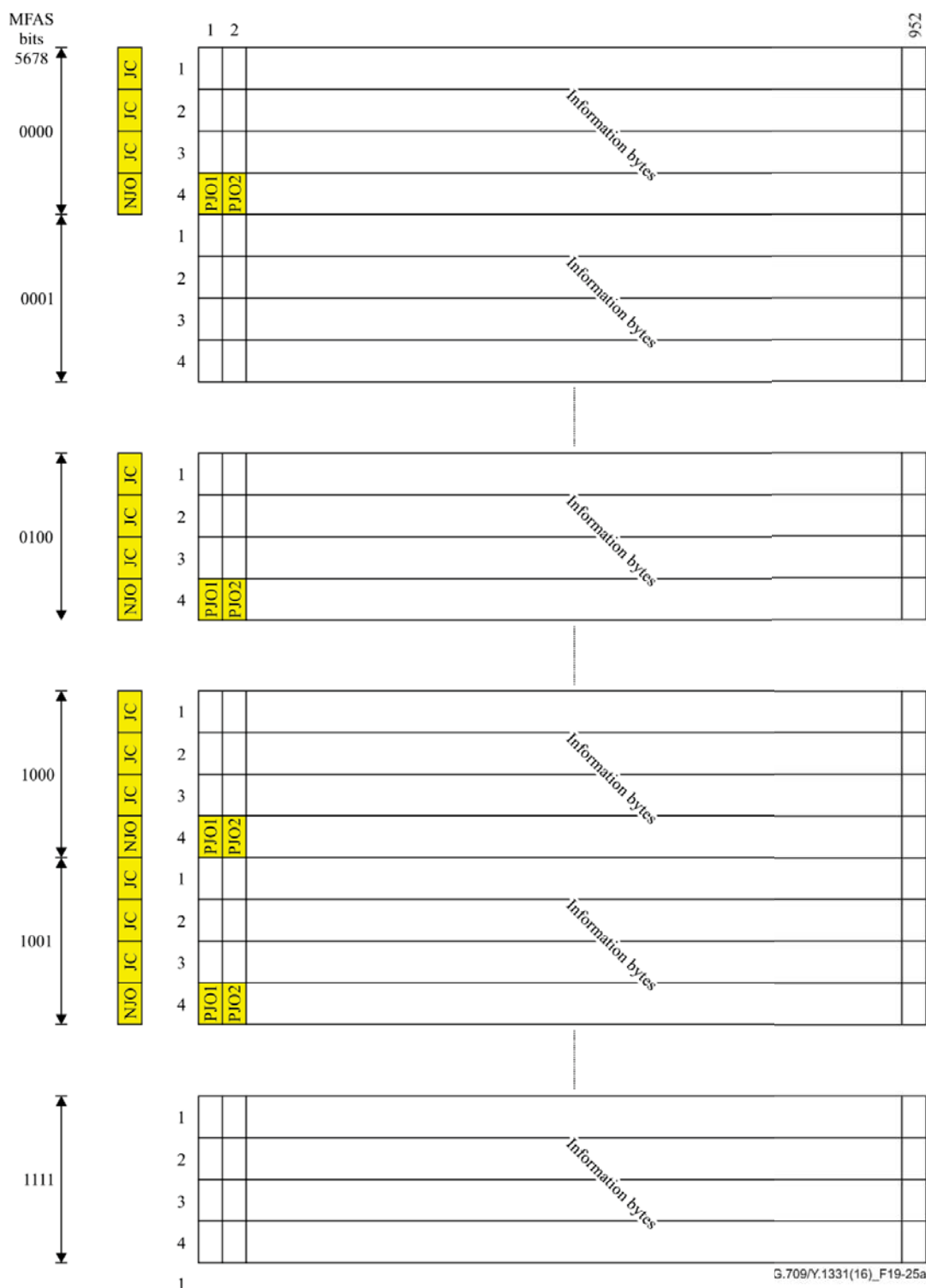
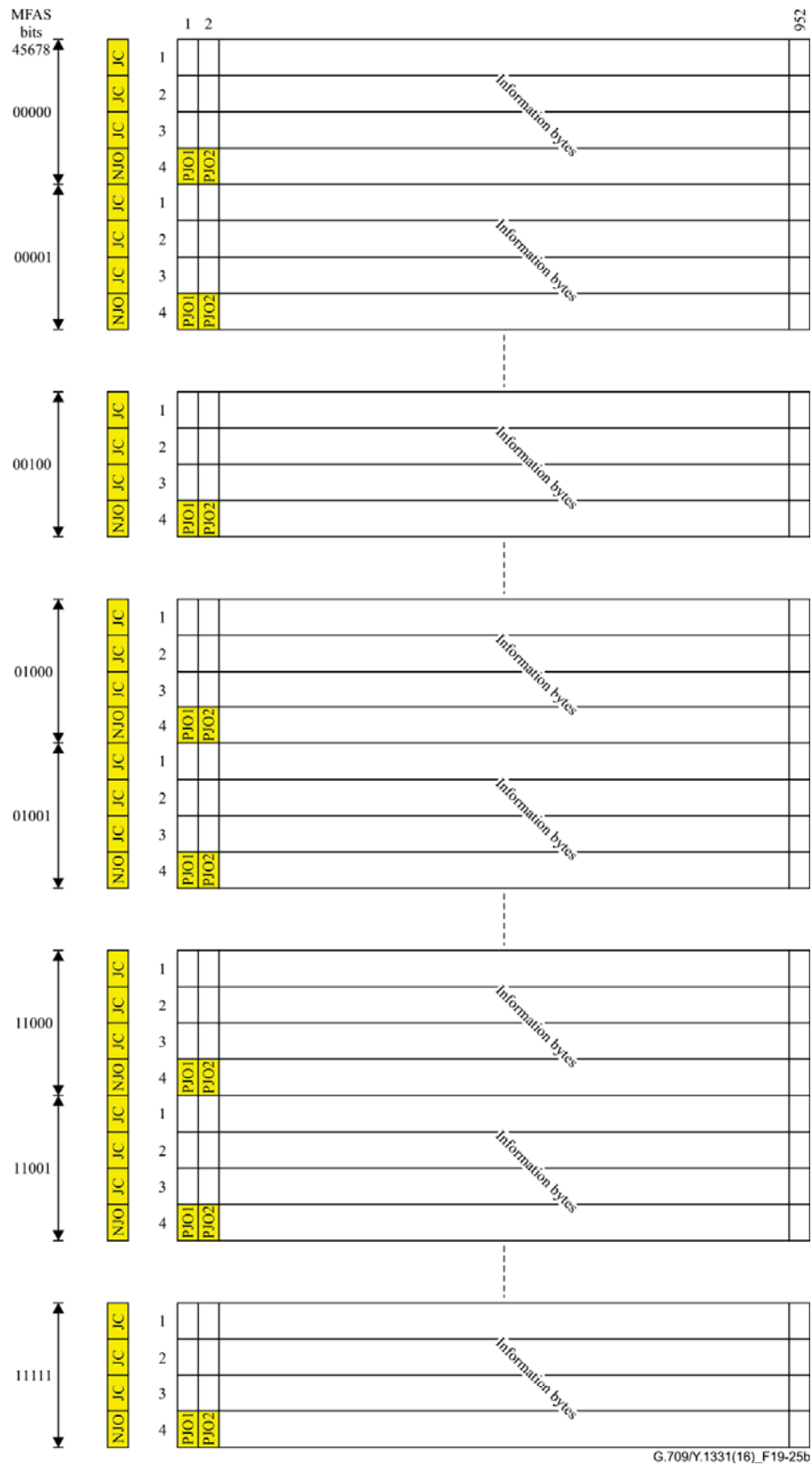


Figure 19-25A – ODTU23 frame format and mapping of ODU2 (mapping in 2.5G TS 1,5,9,10)



G.709/Y.1331(16)_F19-25b

**Figure 19-25B – ODTU23 frame format and mapping of ODU2
(mapping in 1.25G TS 1,2,5,9,10,25,26,32)**

For the case where the ODU_j is received from the output of a fabric (ODU connection function), the incoming signal may contain (in the case of an open matrix connection) the ODU_j-OCI signal as specified in clause 16.5.2. This ODU_j-OCI signal is then mapped into the ODTU_k.M.

NOTE 1 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such a unit is intentionally absent (i.e., not installed), the associated ODTU_k.M signals should carry an ODU_j-OCI signal. If such a unit is installed but temporarily removed as part of a repair action, the associated ODTU_k.M signal should carry an ODU_j-AIS signal.

A group of 'M' successive extended ODU_j bytes is de-mapped from a group of 'M' successive ODTU_k.M bytes.

NOTE 2 – For the case where the ODU_j signal is output as an OTU_j signal, frame alignment of the extracted extended ODU_j signal is to be recovered to allow frame synchronous mapping of the ODU_j into the OTU_j signal.

During a signal fail condition of the incoming ODU_k/OPU_k signal (e.g., in the case of an ODU_k-AIS, ODU_k-LCK, ODU_k-OCI condition) the ODU_j-AIS pattern as specified in clause 16.5.1 is generated as a replacement signal for the lost ODU_j signal.

The values of M, m, C_{m,min}, C_{m,max}, n, C_{n,min} and C_{n,max} for ODU_j into ODTU_k.ts are as follows:

$$M = \text{ceiling} \left(\frac{ODUj_nom_bit_rate \times (1 + ODUj_bit_rate_tolerance + 0.00006)}{(ODTuk1_nom_bit_rate \times (1 - ODTuks_bit_rate_tolerance))} \right) \text{ for ODU}_j \text{ with } j \neq \text{flex(GFP)} \quad (19-1a)$$

$$M = ODUk_bit_rate / ODUks_bit_rate \text{ for ODU}_j \text{ with } j = \text{flex(GFP)} \quad (19-1b)$$

$$m = 8 \times M \quad (19-2)$$

$$c_{m,nom} = \left(\frac{ODUj_nom_bit_rate \times \text{Number_of_GMP_blocks_in_ODTUKts}}{ODTuk1_nom_bit_rate \times M} \right) \quad (19-3)$$

$$c_{m,min} = c_{m,nom} \times \left(\frac{1 - ODUj_bit_rate_tolerance}{1 + ODTuk1_bit_rate_tolerance} \right) \quad (19-4)$$

$$c_{m,max} = c_{m,nom} \times \left(\frac{1 + ODUj_bit_rate_tolerance}{1 - ODTuk1_bit_rate_tolerance} \right) \quad (19-5)$$

$$C_{m,min} = \text{floor}(c_{m,min}) \quad (19-6)$$

$$C_{m,max} = \text{ceiling}(c_{m,min}) \quad (19-7)$$

$$n = 8 \quad (19-8)$$

$$c_{n,nom} = \left(\frac{ODUj_nom_bit_rate \times \text{Number_of_GMP_blocks_in_ODTUKts}}{ODTuk1_nom_bit_rate} \right) \quad (19-9)$$

$$c_{n,min} = c_{n,nom} \times \left(\frac{1 - ODUj_bit_rate_tolerance}{1 + ODTuk1_bit_rate_tolerance} \right) \quad (19-10)$$

$$c_{n,max} = c_{n,nom} \times \left(\frac{1 + ODUj_bit_rate_tolerance}{1 - ODTuk1_bit_rate_tolerance} \right) \quad (19-11)$$

$$C_{n,min} = \text{floor}(c_{n,min}) \quad (19-12)$$

$$C_{n,max} = \text{ceiling}(c_{n,min}) \quad (19-13)$$

$C_{m,min}$, $C_{n,min}$ ($n=8$), $C_{m,max}$ and $C_{n,max}$ ($n=8$) values represent the boundaries of ODU_j/ODTU_k.M ppm offset combinations (i.e., min. ODU_j/max. ODTU_k.M and max. ODU_j/min. ODTU_k.M). In steady state, given instances of ODU_k/ODTU_k.M offset combinations should not result in generated C_n and C_m values throughout this range but rather should be within as small a range as possible.

NOTE – Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

19.6.1 Mapping ODU_j into ODTU2.M

Groups of M successive bytes of the extended ODU_j ($j = 0$, flex) signal are mapped into a group of M successive bytes of the ODTU2.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU2.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU2.M payload area are numbered from 1 to 15232.

The ODTU2.M payload byte numbering for GMP M-byte (m-bit) blocks is illustrated in Figure 19-27. In row 1 of the ODTU2.M multiframe the first M-bytes will be labelled 1, the next M-bytes will be labelled 2, etc.

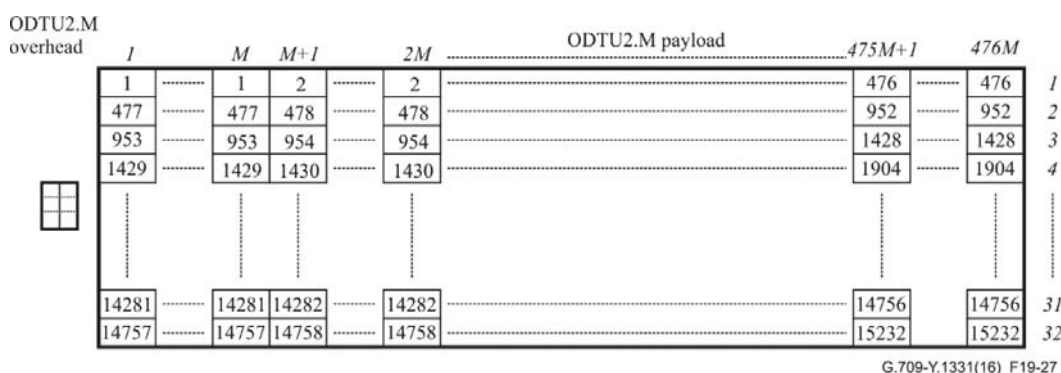


Figure 19-27 – ODTU2.M GMP byte numbering

19.6.2 Mapping ODU_j into ODTU3.M

Groups of M successive bytes of the extended ODU_j ($j = 0$, 2e, flex) signal are mapped into a group of M successive bytes of the ODTU3.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU3.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU3.M payload area are numbered from 1 to 15232.

The ODTU3.M payload byte numbering for GMP M-byte (m-bit) blocks is illustrated in Figure 19-28. In row 1 of the ODTU3.M multiframe the first M-bytes will be labelled 1, the next M-bytes will be labelled 2, etc.

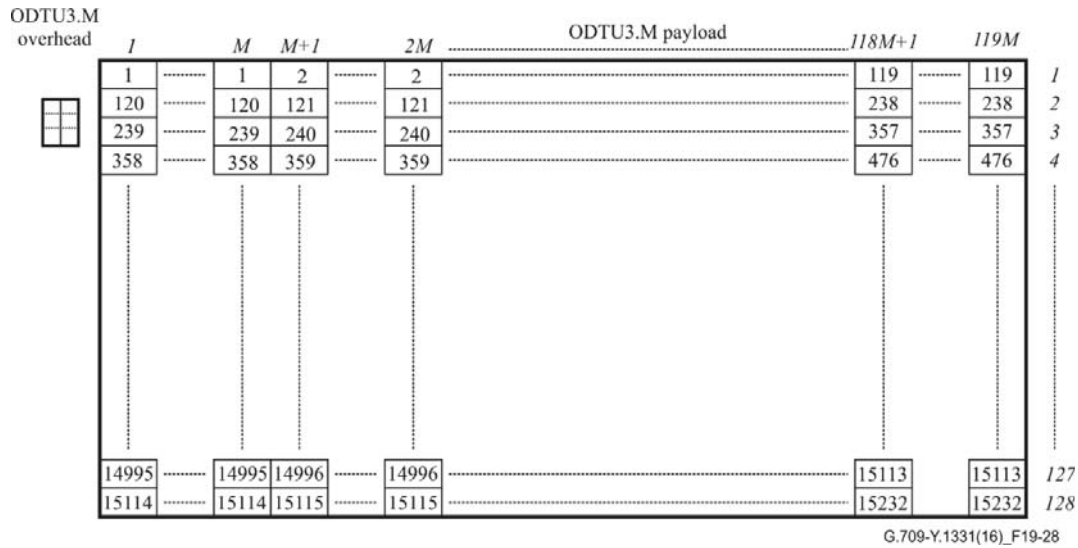


Figure 19-28 – ODTU3.M GMP byte numbering

19.6.3 Mapping ODUj into ODTU4.M

Groups of M successive bytes of the extended ODU $_j$ ($j = 0, 1, 2, 2e, 3, \text{flex}$) signal are mapped into a group of M successive bytes of the ODTU4.M payload area under control of the GMP data/stuff control mechanism. Each group of M bytes in the ODTU4.M payload area may either carry M ODU bytes, or carry M stuff bytes. The value of the stuff bytes is set to all-0s.

The groups of M bytes in the ODTU4.M payload area are numbered from 1 to 15200.

The ODTU4.M payload byte numbering for GMP M -byte (m -bit) blocks is illustrated in Figure 19-29. In row 1 of the ODTU4.M multiframe the first M -bytes will be labelled 1, the next M -bytes will be labelled 2, etc.

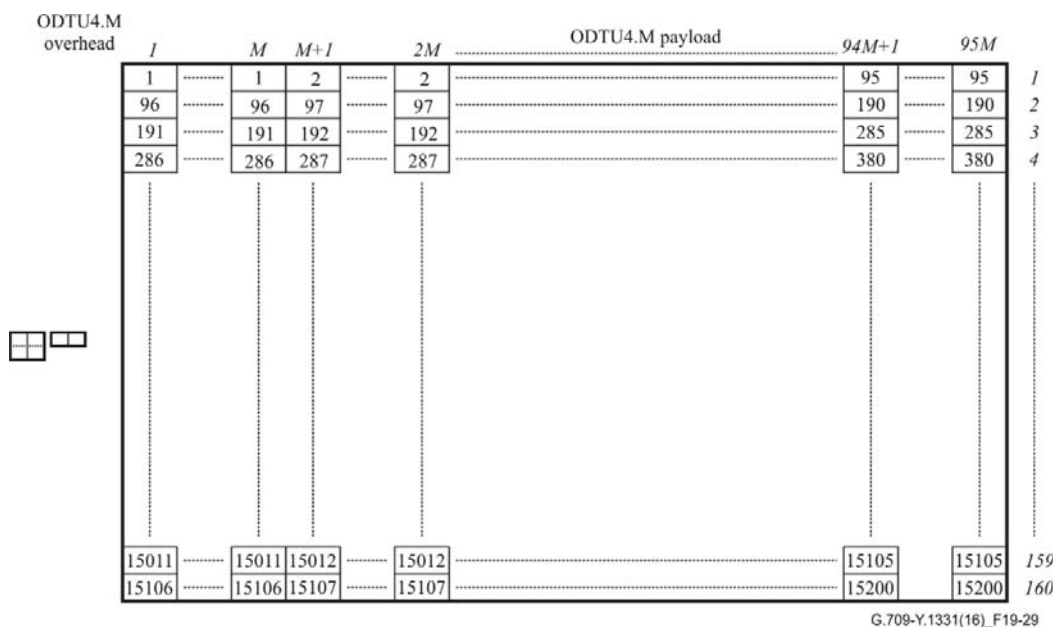


Figure 19-29 – ODTU4.M GMP byte numbering

20 Mapping ODUk signals into the ODTUCn signal and the ODTUCn into the OPUCn tributary slots

This clause specifies the multiplexing of:

- ODUk into OPUCn using a client agnostic generic mapping procedure (GMP).

This ODUk into OPUCn multiplexing is performed in two steps:

- 1) asynchronous mapping of ODUk into optical data tributary unit (ODTUCn) using GMP;
- 2) byte-synchronous mapping of ODTUCn into one or more OPUCn tributary slots.

The OPUCn supports up to 10n different ODUk signals.

20.1 OPUCn tributary slot definition

The OPUCn consists of n OPUC. Each OPUC is divided into 20 tributary slots (TS) and these tributary slots are 16-byte interleaved within the OPUC payload area. A tributary slot includes a part of the OPUC OH area and a part of the OPUC payload area. The bytes of the ODUk frame are mapped into the ODTUCn payload area and the ODTUCn bytes are mapped into the OPUCn tributary slot or slots. The bytes of the ODTUCn justification overhead are mapped into the OPUCn OH area.

There is *only* one type of tributary slot:

- 1) Tributary slot with a bandwidth of approximately 5 Gbit/s; an OPUCn is divided into 20n tributary slots, numbered 1.1 to n.20.

20.1.1 OPUCn tributary slot allocation

Figures 20-1 and 20-2 present the OPUC 5G tributary slot allocation. An OPUC is divided into 20 5G tributary slots (named TS #A.B where A = 1...n which denotes the number of the OPUC within the OPUCn and B = 1...20 which denotes the number of the TS within the OPUC), which are located in columns 17 to 3824. The OPUC multi-frame may be represented in an 80 row by 3810 column format (Figure 20-1) and in a 8 row by 38100 column format (Figure 20-2).

An OPUC 5G tributary slot occupies 5% of the OPUC payload area. It is a structure with 119 16-byte columns by 8(20×4/10) rows (see Figure 20-2) plus a tributary slot overhead (TSOH). The 20 OPUC 5G TSs are 16-byte interleaved in the OPUC payload area and the 20 OPUC TSOHs are frame interleaved in the OPUC overhead area.

The tributary slot overhead (TSOH) of OPUC tributary slots is located in rows 1 to 3, columns 15 and 16 of the OPUC frame.

The TSOH for a 5G tributary slot is available once every 20 frames. A 20-frame multiframe structure is used for this assignment. This multiframe structure is locked to bits 4, 5, 6, 7 and 8 of the OMFI byte as shown in Table 20-1 and Figure 20-1.

Figure 20-3 presents the 5G tributary slots in the OPUCn for the case of 16-byte interleaving of the OPUC instances. This interleaving presents the tributary slot order within the OPUCn.

OMFI bits	Row	Column								
		15-16	17-32	33-48	49-320	321-336	337-352	353-368	369-3807	3808-3824
45678	1	TSOH TS #A.1	TSA.1	TSA.2	:	TSA.20	TSA.1	TSA.2	:	TSA.18
	2		TSA.19	TSA.20	:	TSA.18	TSA.19	TSA.20	:	TSA.16
	3		TSA.17	TSA.18	:	TSA.16	TSA.17	TSA.18	:	TSA.14
	4	PSI OMFI	TSA.15	TSA.16	:	TSA.14	TSA.15	TSA.16	:	TSA.12
00000	1	TSOH TS #A.2	TSA.13	TSA.14	:	TSA.12	TSA.13	TSA.14	:	TSA.10
	2		TSA.11	TSA.12	:	TSA.10	TSA.11	TSA.12	:	TSA.8
	3		TSA.9	TSA.10	:	TSA.8	TSA.9	TSA.10	:	TSA.6
	4	PSI OMFI	TSA.7	TSA.8	:	TSA.6	TSA.7	TSA.8	:	TSA.4
00010	1	TSOH TS #A.3	TSA.5	TSA.6	:	TSA.4	TSA.5	TSA.6	:	TSA.2
	2		TSA.3	TSA.4	:	TSA.2	TSA.3	TSA.4	:	TSA.20
	3		TSA.1	TSA.2	:	TSA.20	TSA.1	TSA.2	:	TSA.18
	4	PSI OMFI	TSA.19	TSA.20	:	TSA.18	TSA.19	TSA.20	:	TSA.16
.....	1								
	2								
	3								
	4								
10011	1	TSOH TS #A.20	TSA.9	TSA.10	:	TSA.8	TSA.9	TSA.10	:	TSA.6
	2		TSA.7	TSA.8	:	TSA.6	TSA.7	TSA.8	:	TSA.4
	3		TSA.5	TSA.6	:	TSA.4	TSA.5	TSA.6	:	TSA.2
	4	PSI OMFI	TSA.3	TSA.4	:	TSA.2	TSA.3	TSA.4	:	TSA.20

Figure 20-1 – OPUC tributary slot allocation in 80 row x 3810 column format

Multi-frame Row	Column Frame	1	2	3	4	5	6	7	8
1	12, 3(1 st 1/2)	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
2	3(2 nd 1/2), 4, 5	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
3	6, 7, 8(1 st 1/2)	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
4	8(2 nd 1/2), 9, 10	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
5	11, 12, 13(1 st 1/2)	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
6	13(2 nd 1/2), 14, 15	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
7	16, 17, 18(1 st 1/2)	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH
8	18(2 nd 1/2), 19, 20	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH	TS OH

Figure 20-2 – OPUC tributary slot allocation in 8 row x 38100 column format

150

Table 20-1 – OPUCn tributary slot OH allocation

OMFI bits 4 5 6 7 8	TSOH 5G TS
0 0 0 0 0	A.1
0 0 0 0 1	A.2
0 0 0 1 0	A.3
0 0 0 1 1	A.4
0 0 1 0 0	A.5
0 0 1 0 1	A.6
0 0 1 1 0	A.7
0 0 1 1 1	A.8
0 1 0 0 0	A.9
0 1 0 0 1	A.10
0 1 0 1 0	A.11
0 1 0 1 1	A.12
0 1 1 0 0	A.13
0 1 1 0 1	A.14
0 1 1 1 0	A.15
0 1 1 1 1	A.16
1 0 0 0 0	A.17
1 0 0 0 1	A.18
1 0 0 1 0	A.19
1 0 0 1 1	A.20

20.2 ODTUCn definition

The optical data tributary unit *Cn* (ODTUCn) carries a justified ODUk signal. There is one type of ODTUCn:

- ODTUCn.ts (ts = 1 to 20n) in which a ODUk (k = 0,1,2e,3,4,flex) signal is mapped via the generic mapping procedure (GMP) defined in clause 20.5.

Optical data tributary unit Cn.ts (ODTUCn.ts)

The optical data tributary unit Cn.ts (ODTUCn.ts) is a structure which consists of an ODTUCn.ts payload area and an ODTUCn.ts overhead area (Figure 20-4). The ODTUCn.ts payload area has 119×ts (ts=1 to 20n) 16-byte-columns and 8 rows (15232×ts bytes) and the ODTUCn.ts overhead area has one times 6 bytes. The ODTUCn.ts is carried in "ts" 5G tributary slots of an OPUCn.

The location of the ODTUCn.ts overhead depends on the OPUCn tributary slot used when multiplexing the ODTUCn.ts in the OPUCn (see clause 20.1.1). The single instance of an ODTUCn.ts overhead is located in the OPUCn TSOH of the last OPUCn tributary slot allocated to the ODTUCn.ts.

The ODTUCn.ts overhead carries the GMP justification overhead as specified in clause 20.4.

The ODTUCn.ts overhead carries the GMP justification overhead consisting of 6 bytes of justification control (JC1, JC2, JC3, JC4, JC5, JC6) which carry the GMP C_m and ΣC_{8D} information.

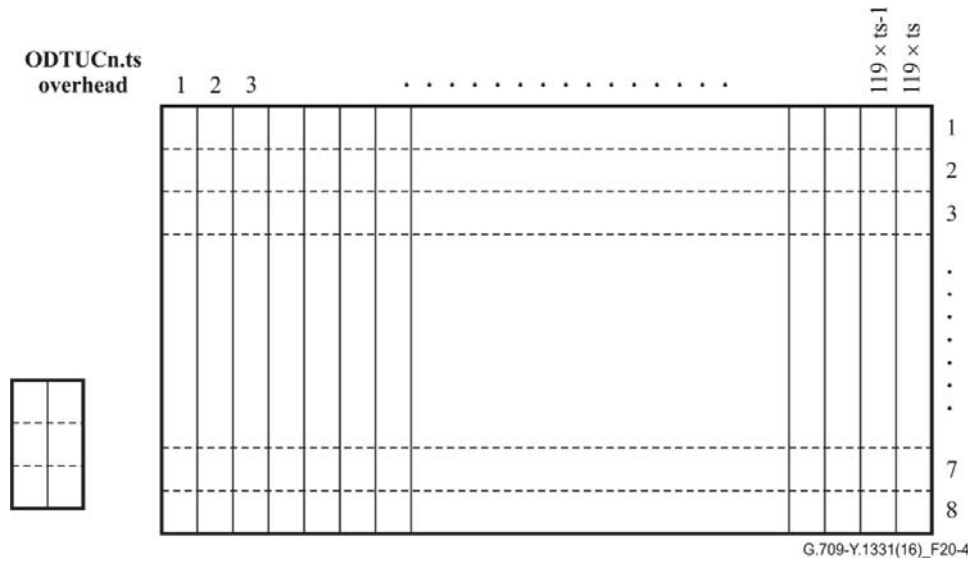


Figure 20-4 – ODTUCn.ts frame formats

20.3 Multiplexing ODTUCn signals into the OPUCn

Multiplexing an ODTUCn.ts signal into an OPUCn is realized by mapping the ODTUCn.ts signal into ts (of $20n$) arbitrary OPUCn 5G tributary slots: OPUCn TS #A₁.B₁, TS #A₂.B₂, .. , TS #A_{ts}.B_{ts} with $1 \leq n*(B_1-1)+A_1 < n*(B_2-1)+A_2 < .. < n*(B_{ts}-1)+A_{ts} \leq 20n$.

NOTE 1 – TS #A₁.B₁, TS #A₂.B₂, ..., TS #A_{ts}.B_{ts} do not have to be sequential in the OPUCn TS sequence; the TSs can be arbitrarily selected to prevent bandwidth fragmentation.

NOTE 2 – TS order is illustrated in Figure 20-5.

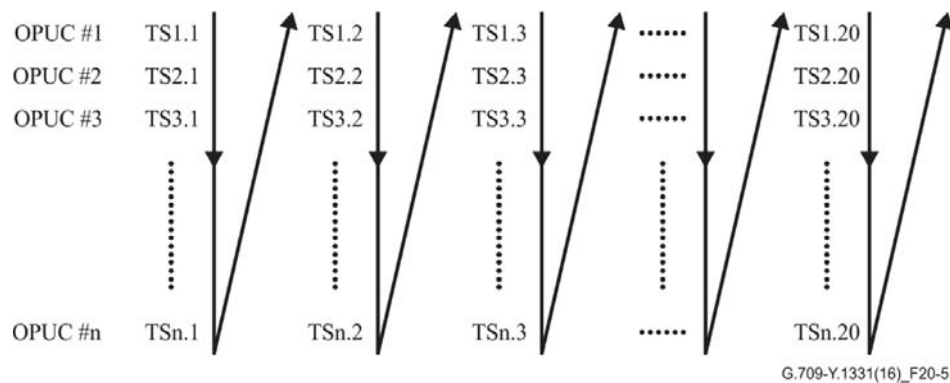


Figure 20-5 – OPUCn TS order

The OPUCn overhead for these multiplexed signals consists of a payload type (PT), the multiplex structure identifier (MSI), the OPUCn multiframe identifier, the OPUCn tributary slot overhead carrying the ODTUCn overhead and depending on the ODTU type one or more bytes reserved for future international standardization.

20.3.1 ODTUCn.ts mapping into ts OPUCn 5G tributary slots

A 16-byte of the ODTUCn.ts payload signal is mapped into a 16-byte of an OPUCn 5G TS #A_i.B_i ($i = 1, ..., ts$) payload area, as indicated in Figure 20-6.

A byte of the $ODTUC_n.ts$ overhead is mapped into a TSOH byte of TS #A_{ts}.B_{ts} within columns 15 and 16, rows 1 to 3 of the last OPUC_n 5G tributary slot allocated to the $ODTUC_n.ts$.

The remaining OPUC_n TSOH bytes are reserved for future international standardization.

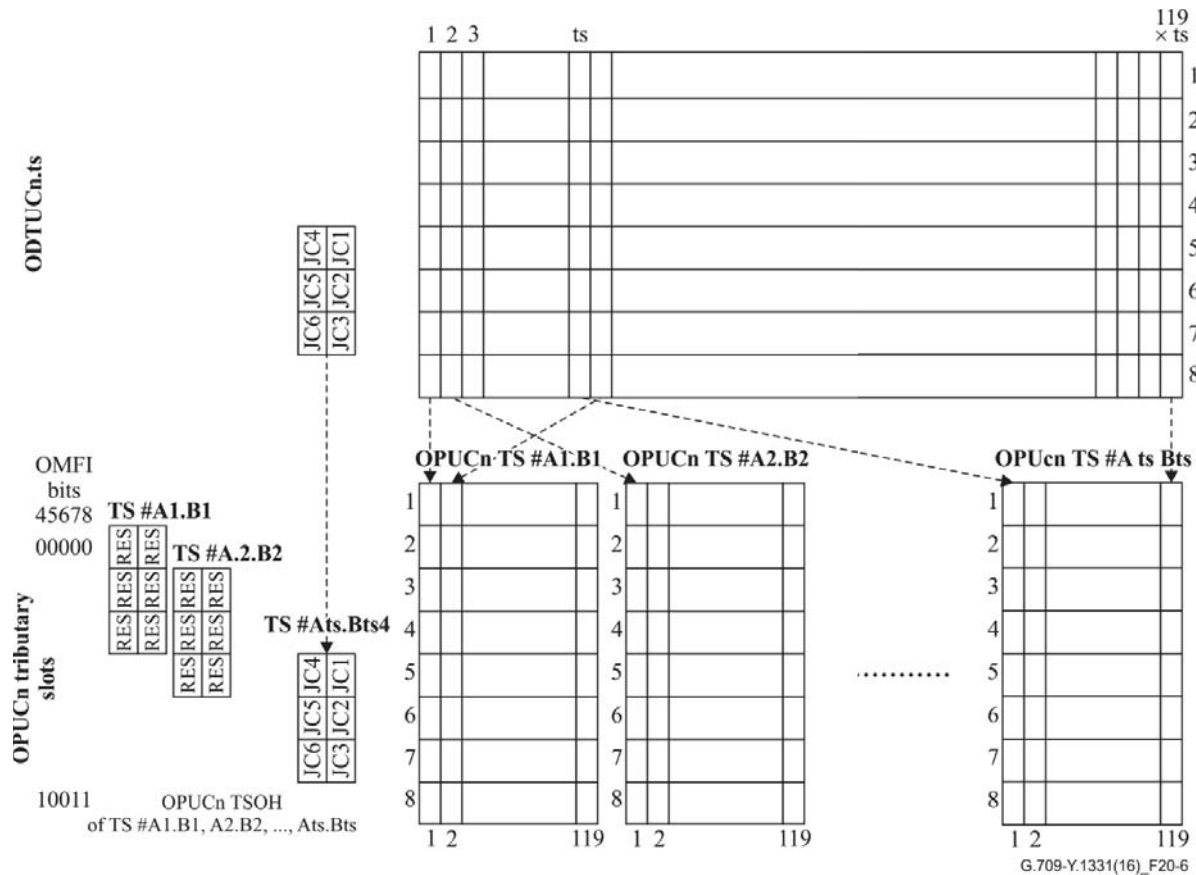


Figure 20-6 – Mapping of $ODTUC_n.ts$ into 'ts' OPUC_n 5G tributary slots

20.4 OPUC_n multiplex overhead and ODTU justification overhead

The OPUC_n multiplex overhead consists of a multiplex structure identifier (MSI), an OPU multiframe identifier (OMFI), an ODTU overhead and bytes reserved for future international standardization.

The OPUC_n MSI, OMFI and RES overhead locations are shown in Figure 20-7.

ODTUC_n.ts overhead

The $ODTUC_n.ts$ overhead carries the GMP justification overhead consisting of 18 bits of justification control (JC1[3-8], JC2[3-8], JC3[3-8]) which carry the 10-bit GMP C_m information and ODU_k ($k=0,1,2,e,3,4,flex$) specific 30 bits of justification control (JC1[1-2], JC2[1-2], JC3[1-2], JC4, JC5, JC6) which carry the 18-bit GMP ΣC_{8D} information.

The JC1, JC2, JC3, JC4, JC5 and JC6 overhead locations are shown in Figure 20-7.

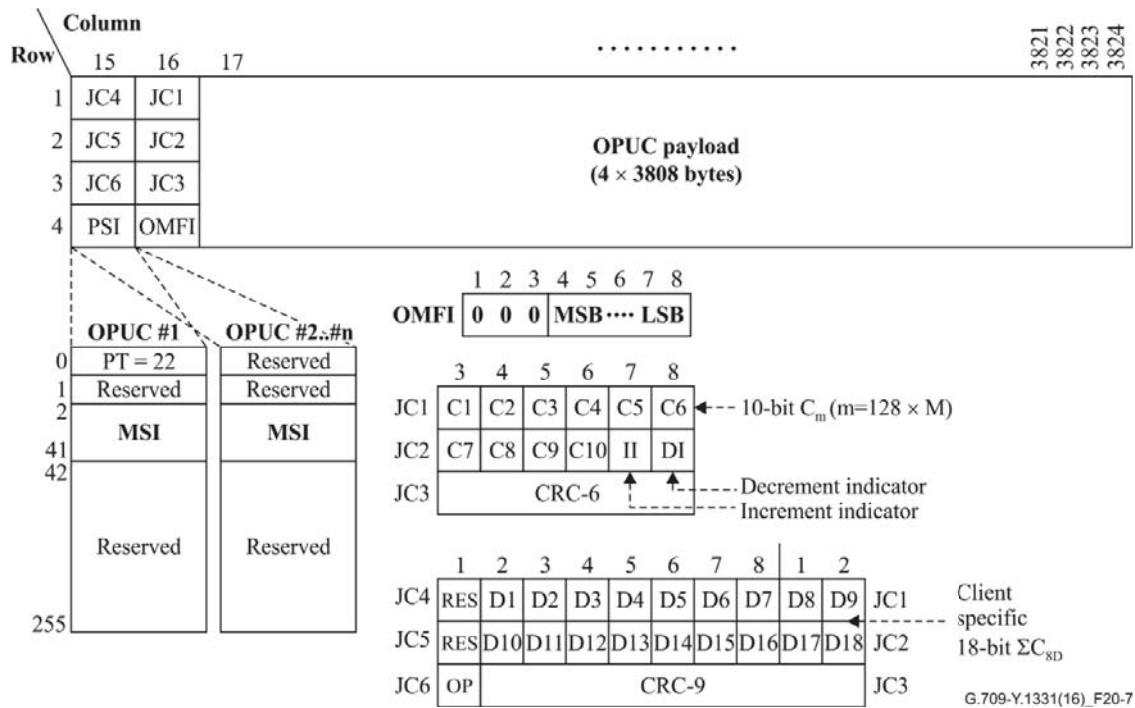


Figure 20-7 – OPUCn multiplex overhead (payload type = 22)

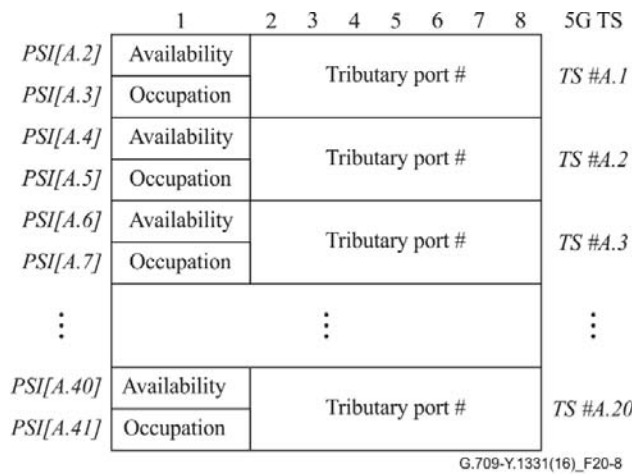
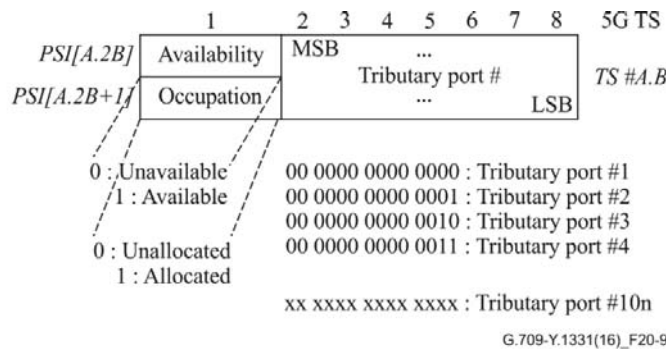
20.4.1 OPUCn multiplex structure identifier (MSI)

The OPUCn multiplex structure identifier (MSI) overhead, which encodes the ODU multiplex structure in the OPU, is located in the mapping specific area of the PSI signal (refer to Figure 20-7 for the MSI location in OPUCn with PT=22). The MSI has a fixed length of 40n bytes and indicates the ODTU content of each tributary slot (TS) of an OPUCn. Two bytes are used for each TS.

20.4.1.1 OPUCn multiplex structure identifier (MSI) – Payload type 22

For the 20n OPUCn 5G tributary slots n times 40 bytes of the PSI are used (named PSI[x.y] where x = 1...n and y = 2...41, that is PSI[1.2], PSI[1.3],...PSI[1.41], PSI[2.2], PSI[2.3],..., PSI[2.41], PSI[3.2],...,PSI[n-1.41], PSI[n.2]..., PSI[n.41]) as MSI bytes as shown in Figure 20-7. The MSI indicates the ODTU content of each tributary slot of an OPU. Two bytes are used for each tributary slot as illustrated in Figure 20-8 and the encoding of the fields of each MSI instance is illustrated in Figure 20-9.

- The TS availability bit 1 indicates if the tributary slot is available or unavailable.
- The TS occupation bit 9 indicates if the tributary slot is allocated or unallocated.
- The tributary port # in bits 2 to 8 and 10 to 16 indicates the port number of the ODTUCn.ts that is being transported in this TS; a flexible assignment of tributary port to tributary slots is possible. ODTUC.ts tributary ports are numbered 1 to 10n. The value is set to all-0s when the occupation bit has the value 0 (tributary slot is unallocated).

NOTE – $A = 1 \dots n$ **Figure 20-8 – OPUCn 5G TS MSI coding – Payload type 22**NOTE – $A = 1 \dots n, B = 1 \dots 20$ **Figure 20-9 – OPUCn MSI coding – Payload type 22****20.4.2 OPUCn payload structure identifier reserved overhead (RES)**

216*n-1 (OPUCn) bytes are reserved in the OPUCn PSI for future international standardization. These bytes are located in $PSI[x.1]$ and $PSI[x.42]$ to $PSI[x.255]$ of the OPUCn overhead where $x = 1 \dots n$ and $PSI[x.0]$, $x=2$ to n . These bytes are set to all-0s.

20.4.3 OPUCn multiplex justification overhead (JOH)

The generic mapping procedure (GMP) is used for the mapping of an ODU k into ODTUC $n.ts$. GMP uses ODU k and OPUC n independent stuff and justification opportunity definitions (ODTUC $n.ts$). Stuff locations within an ODTUC $n.ts$ are determined by means of a formula which is specified in clause 20.4.3.1.

20.4.3.1 Generic mapping procedure (GMP)

The justification overhead (JOH) for the generic mapping procedure consists of two groups of three bytes of justification control; JC1, JC2, JC3 and JC4, JC5, JC6. Refer to Figure 20-7.

The bits 3 to 8 of the JC1, JC2 and JC3 bytes consist of a 10-bit C_m field (bits C1, C2, ..., C10), a 1-bit increment indicator (II) field, a 1-bit decrement indicator (DI) field and a 6-bit CRC-6 field which contains an error check code over the JC1, JC2 and JC3 bits 3 to 8 fields.

The bits 1 and 2 of the JC1, JC2, JC3 bytes and bits 2 to 8 of the JC4, JC5 and JC6 bytes consist of a 18-bit ΣC_{nD} field (bits D1, D2, ..., D18), a 9-bit CRC-9 field which contains an error check code over bits 2 to 8 in the JC4, JC5 and JC6 fields plus bits 1 and 2 in the JC1, JC2 and JC3 fields, two bits

reserved for hitless adjustment of ODUflex control as specified in ITU-T G.7044 and a 1-bit odd parity (OP) field which contains an error check code over bit 1 in the JC4 and JC5 fields.

The value of 'm' in C_m is $128 \times \text{'ts'}$ (number of tributary slots occupied by the ODTUCn.ts).

The value of 'n' represents the timing granularity of the GMP C_n parameter, which is also present in ΣC_{nD} . The value of n is 8.

The value of C_m controls the distribution of groups of '16ts' ODUk data bytes into groups of '16ts' ODTUCn.ts payload bytes. Refer to clause 20.5 and Annex D for further specification of this process.

The value of ΣC_{nD} provides additional 'n'-bit timing information, which is necessary to control the jitter and wander performance experienced by the ODUk signal.

The value of C_n (i.e., number of client n-bit data entities per OPUCn multiframe) is computed as follows: $C_n(t) = m \times C_m(t) + (\Sigma C_{nD}(t) - \Sigma C_{nD}(t-1))$. Note that the value C_{nD} is effectively an indication of the amount of data in the mapper's virtual queue that it could not send during that multiframe due to it being less than a 2M-byte word. For the case where the value of ΣC_{nD} in a multiframe 't' is corrupted, it is possible to recover from such error in the next multiframe 't+1'.

20.4.4 OPUCn multiframe identifier overhead (OMFI)

An OPUCn multiframe identifier (OMFI) byte is defined in row 4, column 16 of the OPUC #1 to #n overhead (Figure 20-10). The value of bits 4 to 8 of the OMFI byte will be incremented each OPUCn frame to provide a 20 frame multiframe for the multiplexing of ODUk signals into the OPUCn.

NOTE 1 – It is an option to align the OMFI = 0 position with MFAS = 0 position every 1280 (the least common multiple of 20 and 256) frame periods.

NOTE 2 – OMFI must be copied in all n OPUC instances at the source, and only 1 need to be processed at the sink.

OMFI OH Byte							
1	2	3	4	5	6	7	8
Fixed to 0	Fixed to 0	Fixed to 0	0	0	0	0	0
			0	0	0	0	1
			0	0	0	1	0
			0	0	0	1	1
0			0	1	0	0	
0			0	1	0	1	
0			0	1	1	0	
0			0	1	1	1	
0			1	0	0	0	
0			1	0	0	1	
0			1	0	1	0	
0			1	0	1	1	
0			1	1	0	0	
0			1	1	0	1	
0			1	1	1	0	
0			1	1	1	1	
1			0	0	0	0	
1			0	0	0	1	
1			0	0	1	0	
1			0	0	1	1	
0			0	0	0	0	
0			0	0	0	1	

OMFI sequence

Figure 20-10– OPUCn multiframe identifier (OMFI) overhead

20.5 Mapping ODU_k into ODTUC_n.ts

The mapping of ODU_k ($k = 0, 1, 2, 2e, 3, 4$, flex) signals (with up to ± 100 ppm bit-rate tolerance) into the ODTUC_n.ts ($ts = M$) signal is performed by means of a generic mapping procedure as specified in Annex D.

The OPU_{Cn} and therefore the ODTUC_n.ts signals are created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the ODU_k signal.

The ODU_k signal is extended with a frame alignment overhead as specified in clauses 15.6.2.1 and 15.6.2.2 and an all-0s pattern in the OTU_k overhead field (see Figure 19-22, read "j" by "k").

The extended ODU_k signal is adapted to the locally generated OPU_{Cn}/ODTUC_n.ts clock by means of a generic mapping procedure (GMP) as specified in Annex D. The value of n in c_n and $C_n(t)$ and $C_{nD}(t)$ is specified in Annex D. The value of M is the number of tributary slots occupied by the ODU_k; ODTUC_n.ts = ODTUC_n.M.

A group of M successive extended ODU_k 16-byte (128-bit) words is mapped into a group of M successive ODTUC_n.M 16-byte (128-bit) words.

NOTE 1 – The 16-byte word alignment of the extended ODU_k is preserved through the mapping procedure; e.g., the position of the first 16 OH bytes of the ODU_k is always located after an integer number of 16-byte words from the start of the ODTUC_n.M structure.

The generic mapping process generates for the case of ODU_k signals once per ODTUC_n.M multiframe the $C_m(t)$ and $C_{nD}(t)$ information according to Annex D and encodes this information in the ODTUC_n.ts justification control overhead JC1/JC2/JC3 and JC4/JC5/JC6. The de-mapping process decodes $C_m(t)$ and $C_{nD}(t)$ from JC1/JC2/JC3 and JC4/JC5/JC6 and interprets $C_m(t)$ and $C_{nD}(t)$ according to Annex D. CRC-6 shall be used to protect against an error in bits 3 to 8 of the JC1, JC2, JC3 signals. CRC-9 and Odd Parity shall be used to protect against an error in bits 1 and 2 of JC1, JC2, JC3 and bits 1 to 8 of JC4, JC5, JC6 signals.

During a signal fail condition of the incoming ODU_k signal, this failed incoming signal will contain the ODU_k-AIS signal as specified in clause 16.5.1. This ODU_k-AIS is then mapped into the ODTUC_n.M.

For the case where the ODU_k is received from the output of a fabric (ODU connection function), the incoming signal may contain (in the case of an open matrix connection) the ODU_k-OCI signal as specified in clause 16.5.2. This ODU_k-OCI signal is then mapped into the ODTUC_n.M.

NOTE 2 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such a unit is intentionally absent (i.e., not installed), the associated ODTUC_n.M signals should carry an ODU_k-OCI signal. If such a unit is installed but temporarily removed as part of a repair action, the associated ODTUC_n.M signal should carry an ODU_k-AIS signal.

A group of M successive extended ODU_k 16-byte words is de-mapped from a group of M successive ODTUC_n.M 16-byte blocks.

NOTE 3 – For the case where the ODU_k signal is output as an OTU_k signal, frame alignment of the extracted extended ODU_k signal is to be recovered to allow frame synchronous mapping of the ODU_k into the OTU_k signal.

During a signal fail condition of the incoming ODU_{Cn}/OPU_{Cn} signal (e.g., in the case of an ODU_{Cn}-AIS condition) the ODU_k-AIS pattern as specified in clause 16.5.1 is generated as a replacement signal for the lost ODU_k signal.

The values of M , m , $C_{m,min}$, $C_{m,max}$, n , $C_{n,min}$ and $C_{n,max}$ for ODU_k into ODTUC_n.ts are as follows:

$$M = \text{ceiling} \left(\frac{ODUk_nom_bit_rate \times (1 + ODUk_bit_rate_tolerance + 0.00006)}{(ODTUCn1_nom_bit_rate \times (1 - ODTUCnts_bit_rate_tolerance))} \right) \text{ for ODUk with } k \neq \text{flex(GFP)} \quad (20-1a)$$

$$M = \text{ceiling} \left(\frac{\text{ODUk_bit_rate}}{\text{ODUk.ts_bit_rate} \times 4} \right) \text{ for ODUk with } k = \text{flex(GFP)} \quad (20-1b)$$

$$m = 128 \times M \quad (20-2)$$

$$c_{m,nom} = \left(\frac{\text{ODUk_nom_bit_rate} \times \text{Number_of_GMP_blocks_in_ODTUCnts}}{\text{ODTUCn.1_nom_bit_rate} \times M} \right) \quad (20-3)$$

$$c_{m,min} = c_{m,nom} \times \left(\frac{1 - \text{ODUk_bit_rate_tolerance}}{1 + \text{ODTUCn.1_bit_rate_tolerance}} \right) \quad (20-4)$$

$$c_{m,max} = c_{m,nom} \times \left(\frac{1 + \text{ODUk_bit_rate_tolerance}}{1 - \text{ODTUCn.1_bit_rate_tolerance}} \right) \quad (20-5)$$

$$C_{m,min} = \text{floor}(c_{m,min}) \quad (20-6)$$

$$C_{m,max} = \text{ceiling}(c_{m,min}) \quad (20-7)$$

$$n = 8 \quad (20-8)$$

$$c_{n,nom} = \left(\frac{\text{ODUk_nom_bit_rate} \times \text{Number_of_GMP_blocks_in_ODTUCnts} \times 16}{\text{ODTUCn.1_nom_bit_rate}} \right) \quad (20-9)$$

$$c_{n,min} = c_{n,nom} \times \left(\frac{1 - \text{ODUk_bit_rate_tolerance}}{1 + \text{ODTUCn.1_bit_rate_tolerance}} \right) \quad (20-10)$$

$$c_{n,max} = c_{n,nom} \times \left(\frac{1 + \text{ODUk_bit_rate_tolerance}}{1 - \text{ODTUCn.1_bit_rate_tolerance}} \right) \quad (20-11)$$

$$C_{n,min} = \text{floor}(c_{n,min}) \quad (20-12)$$

$$C_{n,max} = \text{ceiling}(c_{n,min}) \quad (20-13)$$

$C_{m,min}$, $C_{n,min}$ ($n=8$), $C_{m,max}$ and $C_{n,max}$ ($n=8$) values represent the boundaries of ODUk/ODTUCn.M ppm offset combinations (i.e., min. ODUk/max. ODTUCn.M and max. ODUk/min. ODTUCn.M). In steady state, given instances of ODUk/ODTUCn.M offset combinations should not result in generated C_n and C_m values throughout this range but rather should be within as small a range as possible.

NOTE – Under transient ppm offset conditions (e.g., AIS to normal signal), it is possible that C_n and C_m values outside the range $C_{n,min}$ to $C_{n,max}$ and $C_{m,min}$ to $C_{m,max}$ may be generated and a GMP de-mapper should be tolerant of such occurrences. Refer to Annex D for a general description of the GMP principles.

20.5.1 Mapping ODUk into ODTUCn.M

Groups of M successive 16-byte words of the extended ODUk ($k = 0, 1, 2, 2e, 3, 4, \text{flex}$) signal are mapped into a group of M successive 16-byte blocks of the ODTUCn.M payload area under control of the GMP data/stuff control mechanism. Each group of M 16-byte blocks in the ODTUCn.M payload area may either carry M ODUk 16-byte words, or carry M stuff 16-byte words. The value of the stuff bytes is set to all-0s.

The groups of M 16-byte blocks in the ODTUCn.M payload area are numbered from 1 to 952.

The ODTUCn.M payload 16-byte numbering for GMP M 16-byte (m -bit) blocks is illustrated in Figure 20-11. In row 1 of the ODTUCn.M multiframe the first M 16-byte blocks will be labeled 1, the next M 16-byte blocks will be labeled 2, etc.

		I	M		$M+1$	M	$118*M+1$	$119*M$			
		1	1	2	2	119	119	I
		120	120	121	121	238	238	2
JC4	JC1	239	239	240	240	357	357	3
JC5	JC2	358	358	359	359	476	476	4
JC6	JC3	477	477	478	478	595	595	5
		596	596	597	597	714	714	6
		715	715	716	716	833	833	7
		834	834	835	835	952	952	8

G.709-Y.1331(16)_F20-11

Figure 20-11 – ODTUC $n.M$ GMP 16-byte block numbering

Annex A

Forward error correction using 16-byte interleaved RS(255,239) codecs

(This annex forms an integral part of this Recommendation.)

The forward error correction for the OTU-k uses 16-byte interleaved codecs using a Reed-Solomon RS(255,239) code. The RS(255,239) code is a non-binary code (the FEC algorithm operates on byte symbols) and belongs to the family of systematic linear cyclic block codes.

For FEC processing, an OTU row is separated into 16 sub-rows using byte-interleaving as shown in Figure A.1. Each FEC encoder/decoder processes one of these sub-rows. The FEC parity check bytes are calculated over the information bytes 1 to 239 of each sub-row and transmitted in bytes 240 to 255 of the same sub-row.

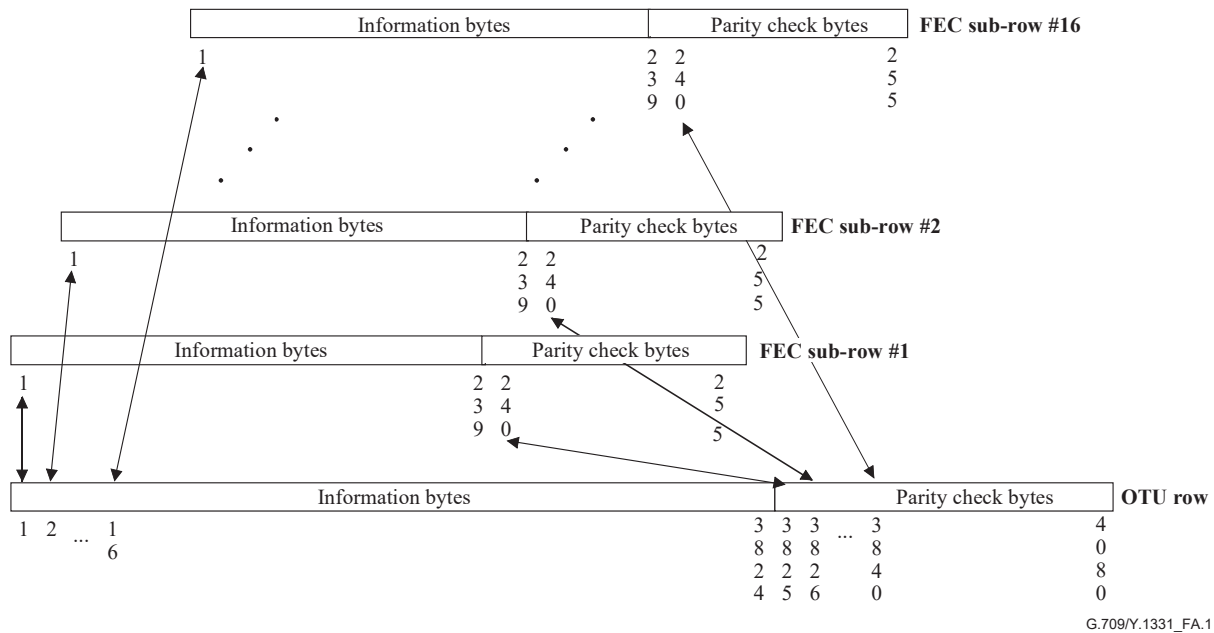


Figure A.1 – FEC sub-rows

The bytes in an OTU row belonging to FEC sub-row X are defined by: $X + 16 \times (i - 1)$ (for $i = 1 \dots 255$).

The generator polynomial of the code is given by:

$$G(z) = \prod_{i=0}^{15} (z - \alpha^i)$$

where α is a root of the binary primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

The FEC code word (see Figure A.2) consists of information bytes and parity bytes (FEC redundancy) and is represented by the polynomial:

$$C(z) = I(z) + R(z)$$

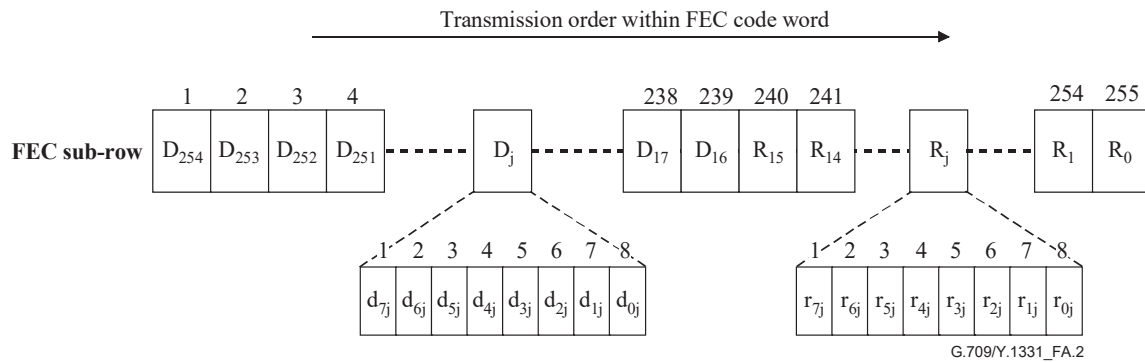


Figure A.2 – FEC code word

Information bytes are represented by:

$$I(z) = D_{254} \cdot z^{254} + D_{253} \cdot z^{253} + \dots + D_{16} \cdot z^{16}$$

Where D_j ($j = 16$ to 254) is the information byte represented by an element out of GF(256) and:

$$D_j = d_{7j} \cdot \alpha^7 + d_{6j} \cdot \alpha^6 + \dots + d_{1j} \cdot \alpha^1 + d_{0j}$$

Bit d_{7j} is the MSB and d_{0j} the LSB of the information byte.

D_{254} corresponds to byte 1 in the FEC sub-row and D_{16} to byte 239.

Parity bytes are represented by:

$$R(z) = R_{15} \cdot z^{15} + R_{14} \cdot z^{14} + \dots + R_1 \cdot z^1 + R_0$$

Where R_j ($j = 0$ to 15) is the parity byte represented by an element out of GF(256) and:

$$R_j = r_{7j} \cdot \alpha^7 + r_{6j} \cdot \alpha^6 + \dots + r_{1j} \cdot \alpha^1 + r_{0j}$$

Bit r_{7j} is the MSB and r_{0j} the LSB of the parity byte.

R_{15} corresponds to the byte 240 in the FEC sub-row and R_0 to byte 255.

$R(z)$ is calculated by:

$$R(z) = I(z) \bmod G(z)$$

where "mod" is the modulo calculation over the code generator polynomial $G(z)$ with elements out of the GF(256). Each element in GF(256) is defined by the binary primitive polynomial

$$x^8 + x^4 + x^3 + x^2 + 1.$$

The Hamming distance of the RS(255,239) code is $d_{\min} = 17$. The code can correct up to 8 symbol errors in the FEC code word when it is used for error correction. The FEC can detect up to 16 symbol errors in the FEC code word when it is used for error detection capability only.

Annex B

Adapting 64B/66B encoded clients via transcoding into 513B code blocks

(This annex forms an integral part of this Recommendation.)

Clients using 64B/66B coding can be adapted in a codeword and timing transparent mapping via transcoding into 513B code blocks to reduce the bit rate that is required to transport the signal. The resulting 513B blocks can be mapped in one of several ways depending on the requirements of the client and the available bandwidth of the container into which the client is mapped. This mapping can be applied to serial or parallel client interfaces.

B.1 Transmission order

The order of transmission of information in all the diagrams in this annex is first from left to right and then from top to bottom.

B.2 Client frame recovery

For 40GBASE-R and 100GBASE-R clients, framing recovery consists of the recovering 64B/66B block lock per the state diagram in Figure 82-10 of [IEEE 802.3]. For other 64B/66B encoded clients, block lock is achieved as per the state diagram in Figure 49-12 of [IEEE 802.3]. Descrambling is performed as per the process shown in Figure 49-10 of [IEEE 802.3].

Each 66B codeword (after block lock) is one of the following:

- a set of eight data bytes with a sync header of "01";
- a control block (possibly including seven or fewer data octets) beginning with a sync header of "10".

The 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane markers) according to the polynomial $G(x) = 1 + x^{39} + x^{58}$. The 64B/66B PCS receive process will descramble the bits other than (1) the sync header of 66B data and control blocks, and (2) the PCS lane markers.

Figure B.1 illustrates the ordering of 64B/66B code blocks after the completion of the recovering process for an interface.

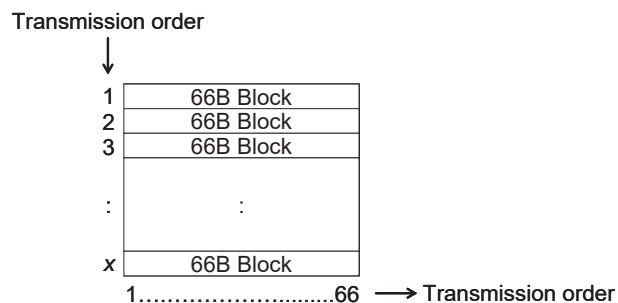


Figure B.1 – Stream of 64B/66B code blocks for transcoding

B.3 Transcoding from 66B blocks to 513B blocks

The transcoding process at the encoder operates on an input sequence of 66B code blocks.

66B control blocks (after descrambling) follow the format shown in Figure B.2.

A group of eight 66B blocks is encoded into a single 513B block. The format is illustrated in Figure B.3.

Input Data			S	Block Payload																																																																																																																																																																
			Y																																																																																																																																																																	
			C																																																																																																																																																																	
Data Block Format			Bit																																																																																																																																																																	
			Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65																																																																																															
DoD1D2D3D4D5D6D7			0	1	D0																D1																D2																D3																D4																D5																D6																D7																																															
Control block formats			Block type field																																																																		4-bit code																																																																																															
CoC1C2C3C4C5C6C7			1	0	0x1e																C0																C1																C2																C3																C4																C5																C6																C7																0001															
CoC1C2C3O4D5D6D7			1	0	0x2d																C0																C1																C2																C3																O4																D5																D6																D7																0010															
CoC1C2C3S4D5D6D7			1	0	0x33																C0																C1																C2																C3																																D5																D6																D7																0111															
OoD1D2D3S4D5D6D7			1	0	0x66																D1																D2																D3																O0																D6																D7																1011																																															
OoD1D2D3O4D5D6D7			1	0	0x55																D1																D2																D3																O0																O4																D6																D7																1101																															
SoD1D2D3D4D5D6D7			1	0	0x78																D1																D2																D3																D4																D5																D6																D7																1110																															
OoD1D2D3C4C5C6C7			1	0	0x4b																D1																D2																D3																O0																C4																C5																C6																C7																1000															
ToC1C2C3C4C5C6C7			1	0	0x87																																C1																C2																C3																C4																C5																C6																C7																0011															
DoT1C2C3C4C5C6C7			1	0	0x99																D0																																C2																C3																C4																C5																C6																C7																0101															
DoD11T2C3C4C5C6C7			1	0	0xaa																D0																D1																C3																C4																C5																C6																C7																1001																															
DoD1D2T3C4C5C6C7			1	0	0xb4																D0																D1																D2																C4																C5																C6																C7																1010																															
DoD1D2D8T4C5C6C7			1	0	0xcc																D0																D1																D2																D3																C5																C6																C7																1100																															
DoD1D2D3D4T5C6C7			1	0	0xd2																D0																D1																D2																D3																D4																C6																C7																0110																															
DoD1D2D3D4D5T6C7			1	0	0xe1																D0																D1																D2																D3																D4																D5																C7																0000																															
DoD1D2D3D4D5D6T7			1	0	0xff																D0																D1																D2																D3																D4																D5																D6																1111																															

Figure B.2 – 66B Block coding

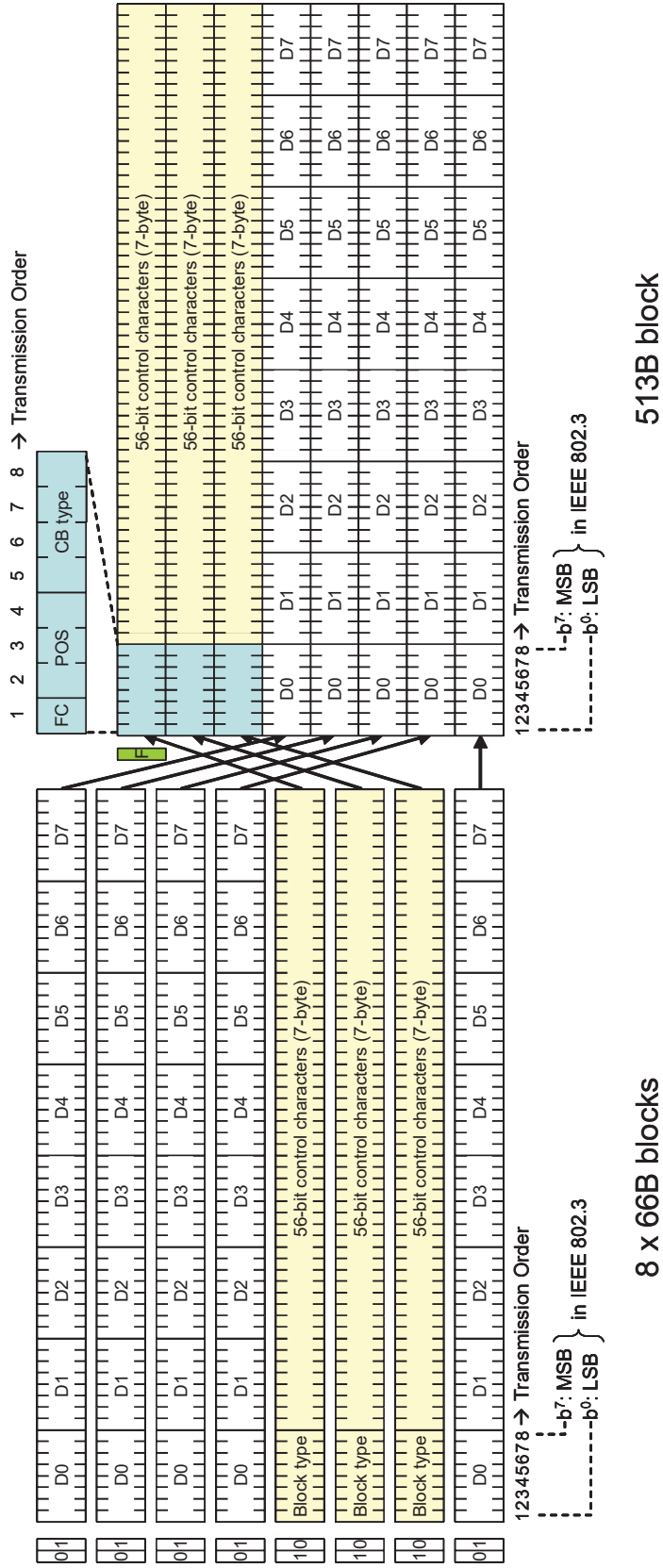


Figure B.3 – 513B block code format

Each of the 66B blocks is encoded into a row of the 8-byte by 8-row structure. Any 66B control blocks (C*B*_{*i*}) are placed into the uppermost rows of the structure in the order received, while any all-data 66B blocks (D*B*_{*i*}) are placed into the lowermost rows of the structure in the order received.

The flag bit "F" is 1 if the 513B structure contains at least one 66B control block, and 0 if the 513B structure contains eight all-data 66B blocks. Because the 66B control blocks are placed into the uppermost rows of the 513B block, if the flag bit "F" is 1, then the first row will contain a mapping of a 66B control block.

A 66B control block is encoded into a row of the structure shown in Figure B.3 as follows: the sync header of "10" is removed. The byte representing the block type field (see Figure B.2) is replaced by the structure shown in Figure B.4:



Figure B.4 – 513B block's control block header

The byte indicating the control block type (one of 15 legal values) is translated into a 4-bit code according to the rightmost column of Figure B.2. The 3-bit POS field is used to encode the position in which this control block was received in the sequence of eight 66B blocks. The flag continuation bit "FC" will be set to a 0 if this is the final 66B control block or PCS lane alignment marker encoded in this 513B block, or to a 1 if one or more 66B control blocks or PCS lane alignment markers follow this one. At the decoder, the flag bit for the 513B block as a whole, plus the flag continuation bits in each row containing the mapping of a 66B control block or PCS lane alignment marker will allow identification of those rows, which can then be restored to their original position amongst any all-data 66B blocks at the egress according to the POS field. The remaining 7 bytes of the row are filled with the last 7 bytes of the 66B control block.

An all-data 66B block is encoded into a row of the 513B block by dropping the sync header and copying the remaining eight bytes into the row. If all eight rows of the 513B block are placements of 66B all-data blocks, the flag bit "F" will be 0. If fewer than eight rows of the 513B block are placements of 66B all-data blocks, they will appear at the end, and the row containing the placement of the final 66B control block will have a flag continuation bit "FC" value of 0.

The decoder operates in the reverse of the encoder to reconstruct the original sequence of 66B blocks. If flag bit "F" is 1, then 66B control blocks starting from the first row of the block are reconstructed and placed in the position indicated by the POS field. This process continues through all of the control blocks working downward from the top row. The final 66B control block placed within the 513B block will be identified when the flag continuation bit "FC" is zero.

The structure of the 512B/513B code block is shown in Figure B.5. For example, if there is a single 64B/66B control block CB1 in a 512B/513B code block and it was originally located between 64B/66B data blocks DB2 and DB3, the first octet of the 64B character will contain 0.010.1101.CB1; the leading bit in the control octet of 0 indicates the flag continuation "FC" that this 64B control block is the last one in the 512B/513B code block, the value of 010 indicates CB1's position "POS" between DB2 and DB3, and the value of 1101 is a four-bit representation of the control code's block type "CB TYPE" (of which the eight-bit original block type is 0x55).

Input client characters	Flag bit	512-bit (64-Octet) field							
All data block	0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
7 data block 1 control block	1	0 AAA aaaa CB1	DB1	DB2	DB3	DB4	DB5	DB6	DB7
6 data block 2 control block	1	1 AAA aaaa CB1	0 BBB bbbb CB2	DB1	DB2	DB3	DB4	DB5	DB6
5 data block 3 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	0 CCC cccc CB3	DB1	DB2	DB3	DB4	DB5
4 data block 4 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	0 DDD dddd CB4	DB1	DB2	DB3	DB4
3 data block 5 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	0 EEE eeee CB5	DB1	DB2	DB3
2 data block 6 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	0 FFF ffff CB6	DB1	DB2
1 data block 7 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	0 GGG gggg CB7	DB1
8 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	1 GGG gggg CB7	0 HHH hhhh CB8

- Leading bit in a 66B control block FC = 1 if there are more than 66B control block and = 0 if this payload contains the last control block in that 513B block
 - AAA = 3-bit representation of the first control code's original position (First control code locator: POS)
 - BBB = 3-bit representation of the second control code's original position (Second control code locator: POS)

 - HHH = 3-bit representation of the eighth control code's original position (Eighth control code locator: POS)
 - aaa = 4-bit representation of the first control code's type (first control block type: CB TYPE)
 - bbb = 4-bit representation of the second control code's type (Second control block type: CB TYPE)

 - hhh = 4-bit representation of the eighth control code's type (Eighth control block type: CB TYPE)
 - CBi = 56-bit representation of the i-th control code characters
 - DBi = 64-bit representation of the i-th data value in order of transmission

G.709-Y.1331(12)_FB.5

Figure B.5 – 513B code block components**B.3.1 Errors detected before 512B/513B encoder**

A set of errors might be detected at the 64B/66B PCS receive process which, in addition to appropriate alarming, needs to send the appropriate signal downstream.

Errors encountered before the encoder, such as loss of client signal, will result in the insertion of an Ethernet LF sequence ordered set prior to this process, which will be transcoded as any other control block. The same action should be taken as a result of failure to achieve 66B block lock on an input signal.

An invalid 66B block will be converted to an error control block before transcoding and the OTN BIP-8 calculation as described in clause E.4.1. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10" and a control block type field which does not appear in Figure B.2. An error control block has sync bits of "10", a block type code of 0x1E, and 8 seven-bit/E/error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

B.3.2 Errors detected by 512B/513B decoder

Several mechanisms will be employed to reduce the probability that the decoder constructs erroneous 64B/66B encoded data at the egress if bit errors have corrupted. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Mechanisms for improving the robustness and for 513B block lock are discussed in Annex F.

B.4 Link fault signalling

In-band link fault signalling in the client 64B/66B code (e.g., if a local fault or remote fault sequence ordered set is being transmitted between Ethernet equipments) is carried transparently according to this transcoding.

Annex C

Adaptation of OTU3 and OTU4 over multichannel parallel interfaces

(This annex forms an integral part of this Recommendation.)

NOTE 1 – This mechanism is designed to allow the use of the optical modules being developed for IEEE 40GBASE-R and 100GBASE-R signals for short-reach client-side OTU3 and OTU4 interfaces respectively. The corresponding physical layer specifications are being added to [ITU-T G.695] and [ITU-T G.959.1].

OTU3 signals may be carried over parallel interfaces consisting of four lanes. This four lane format is referred to as the OTL3.4 format.

OTU4 signals may be carried over parallel interfaces consisting of four or ten lanes, which are formed by bit multiplexing of 20 logical lanes. The four lane format is referred to as the OTL4.4 signal format and the ten lane format is referred to as the OTL4.10 signal format.

NOTE 2 – Ten lane IEEE 100GBASE-R interfaces have no corresponding ITU-T physical layer interface specification.

The OTU3 and OTU4 frames are inversely multiplexed over physical/logical lanes on a 16-byte boundary aligned with the OTUk frame as illustrated in Figure C.1. The OTUk frame is divided into 1020 groups of 16-bytes.

<i>l</i>					<i>4080</i>
<i>l</i>	1:16 (FAS)	17:32	33:48	49:64	4065:4080
2	4081:4096	4097:5012	5013:5028	5029:5044	9145:9160
3	9161:9176	9177:9192	9193:9208	9209:9224	12225:12240
4	12241:12256	12257:12272	12273:12288	12289:13304	16305:16320

Figure C.1 – OTU3 and OTU4 frames divided on 16-byte boundary

OTU3 16-byte increment distribution

Each 16-byte increment of an OTU3 frame is distributed round robin, to each of the four physical lanes. On each OTU3 frame boundary the lane assignments are rotated.

For OTU3, the lane rotation and assignment is determined by the two LSBs of the MFAS as described in Table C.1 and Figure C.2, which indicates the starting group of bytes of the OTU3 frame that are sent on each lane.

NOTE 3 – MFAS is scrambled as defined in clause 11.2.

The pattern repeats every 64 bytes until the end of the OTU3 frame. The following OTU3 frame will use different lane assignments according to the MFAS.

Table C.1 – Lane rotation assignments for OTU3

MFAS 7-8	Lane 0	Lane 1	Lane 2	Lane 3
*00	1:16	17:32	33:48	49:64
*01	49:64	1:16	17:32	33:48
*10	33:48	49:64	1:16	17:32
*11	17:32	33:48	49:64	1:16

The distribution of 16-byte blocks from the sequence of OTU3 frames is illustrated in Figure C.2:

The parallel lanes can be reassembled at the sink by first recovering framing on each of the parallel lanes, then recovering the lane identifiers and then performing lane de-skewing. Frame alignment, lane identifier recovery and multi-lane alignment should operate under 10^{-3} bit error rate conditions before error correction. Refer to [ITU-T G.798] for the specific processing details.

The lane rotation mechanism will place the first 16 bytes of the OTU3 frame on each lane once per 4080×4 (i.e., 16320) bytes (the same as an OTU3 itself). The two LSBs of the MFAS will be the same in each FAS on a particular lane, which allows the lane to be identified. Since the MFAS cycles through 256 distinct values, the lanes can be de-skewed and reassembled by the receiver as long as the total skew does not exceed 127 OTU3 frame periods (approximately 385 μ s). The receiver must use the MFAS to identify each received lane, as lane positions may not be preserved by the optical modules to be used for this application.

OTU4 16-byte increment distribution

Each 16-byte increment of an OTU4 frame is distributed, round robin, to each of the 20 logical lanes. On each OTU4 frame boundary the lane assignments are rotated.

For distribution of OTU4 to twenty logical lanes, since the MFAS is not a multiple of 20, a different marking mechanism must be used. Since the frame alignment signal is 6 bytes (48 bits) and as per [ITU-T G.798] only 32 bits must be checked for frame alignment, the third OA2 byte position will be borrowed as a logical lane marker (LLM). For maximum skew detection range, the lane marker value will increment on successive frames from 0-239 (240 values being the largest multiple of 20 that can be represented in 8-bits). LLM = 0 position shall be aligned with MFAS = 0 position every 3840 (the least common multiple of 240 and 256) frame periods. The logical lane number can be recovered from this value by a modulo 20 operation. Table C.2 and Figure C.3 illustrate how bytes of the OTU4 are distributed in 16-byte increments across the 20 logical lanes.

The pattern repeats every 320 bytes until the end of the OTU4 frame.

The following OTU4 frame will use different lane assignment according to the LLM MOD 20.

Table C.2 – Lane rotation assignments for OTU4

LLM MOD 20	Lane 0	Lane 1	Lane 18	Lane 19
0	1:16	17:32		289:304	305:320
1	305:320	1:16		273:288	289:304
:					
18	33:48	49:64		1:16	17:32
19	17:32	33:48		305:320	1:16

The distribution of 16-byte blocks from the sequence of OTU4 frames is illustrated in Figure C.3.

The parallel lanes can be reassembled at the sink by first recovering framing on each of the parallel lanes, then recovering the lane identifiers and then performing de-skewing of the lanes. Frame alignment, lane identifier recovery and multi-lane alignment should operate under 10^{-3} bit error rate conditions before error correction. Refer to [ITU-T G.798] for specific processing details.

The lane rotation mechanism will place the first 16 bytes of the OTU4 frame on each lane once per 4080×4 (i.e., 16320) bytes (the same as an OTU4 itself). The "LLM MOD 20" will be the same in each FAS on a particular lane, which allows the lane to be identified. Since the LLM cycles through 240 distinct values, the lanes can be de-skewed and reassembled by the receiver as long as the total skew does not exceed 119 OTU4 frame periods (approximately 139 μ s). The receiver must use the

"LLM MOD 20" to identify each received lane, as lane positions may not be preserved by the optical modules to be used for this application.

The lanes are identified, de-skewed, and reassembled into the original OTU4 frame according to the lane marker. The MFAS can be combined with the lane marker to provide additional skew detection range, the maximum being up to the least common multiple "LCM(240, 256)/2 – 1" or 1919 OTU4 frame periods (approximately 2.241 ms). In mapping from lanes back to the OTU4 frame, the sixth byte of each OTU4 frame which was borrowed for lane marking is restored to the value OA2.

Each physical lane of an OTL4.4 carried over a multi-lane SOTU interface is formed by simple bit multiplexing of five logical lanes. At the sink, the bits are disinterleaved into five logical lanes from each physical lane. The sink will identify each logical lane according to the lane marker in the LLM byte. The sink must be able to accept the logical lanes in any position as the ordering of bit multiplexing on each physical lane is arbitrary; the optical module hardware to be used for this application is permitted full flexibility concerning which physical lane will be used for output of each logical lane, and the order of bit multiplexing of logical lanes on each physical output lane.

NOTE 4 – Ten-lane IEEE 100GBASE-R interfaces are specified, although not with ITU-T physical layer specifications. These interfaces may be compatible with a 10-lane interface for OTU4 (OTL4.10), each lane consisting of two bit-multiplexed logical lanes. Refer to [b-ITU-T G-Sup.58].

This mechanism handles any normally framed OTU3 or OTU4 sequence.

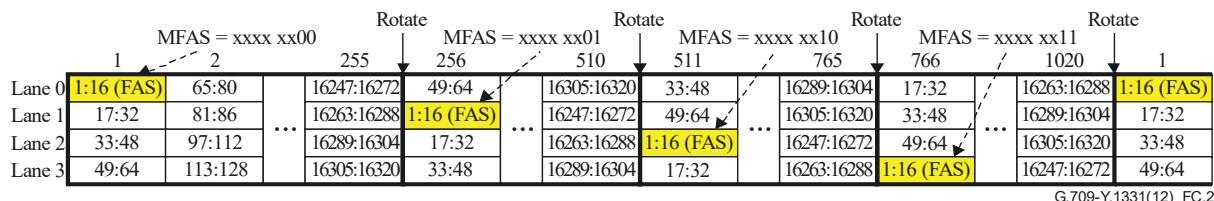


Figure C.2 – Distribution of bytes from OTU3 to parallel lanes

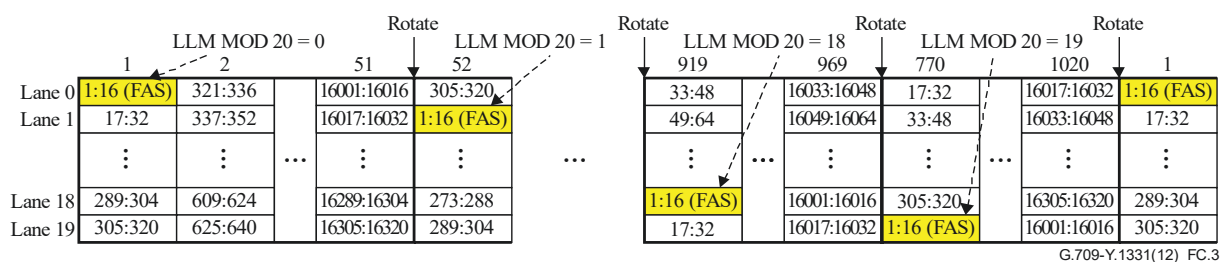


Figure C.3 – Distribution of bytes from OTU4 to parallel lanes

Annex D

Generic mapping procedure principles

(This annex forms an integral part of this Recommendation.)

D.1 Basic principle

For any given CBR client signal, the number of n-bit (e.g., n = 1/8, 1, 8) data entities that arrive during one server frame or server multiframe period is defined by:

$$c_n = \left(\frac{f_{client}}{n} \times T_{server} \right) \quad (D-1)$$

f_{client} : client bit rate

T_{server} : frame period of the server frame or server multiframe

c_n : number of client n-bit data entities per server frame or server multiframe

As only an integer number of n-bit data entities can be transported per server frame or multiframe, the integer value $C_n(t)$ of c_n has to be used. Since it is required that no client information is lost, the rounding process to the integer value has to take care of the truncated part, e.g., a c_n with a value of 10.25 has to be represented by the integer sequence 10,10,10,11.

$$C_n(t) = \text{int} \left(\frac{f_{client}}{n} \times T_{server} \right) \quad (D-2)$$

$C_n(t)$: number of client n-bit data entities per server frame t or server multiframe t (integer)

For the case c_n is not an integer, $C_n(t)$ will vary between:

$$C_n(t) = \text{floor} \left(\frac{f_{client}}{n} \times T_{server} \right) \quad (D-3)$$

and

$$C_n(t) = \text{ceiling} \left(\frac{f_{client}}{n} \times T_{server} \right) = 1 + \text{floor} \left(\frac{f_{client}}{n} \times T_{server} \right) \quad (D-4)$$

The server frame or multiframe rate is defined by the server bit rate and the number of bits per server frame or multiframe:

$$T_{server} = \frac{B_{server}}{f_{server}} \quad (D-5)$$

f_{server} : server bit rate

B_{server} : bits per server frame or multiframe

Combining (D-5) with (D-1) and (D-2) results in:

$$c_n = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{n} \right) \quad (D-6)$$

and

$$C_n(t) = \text{int} \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{n} \right) \quad (\text{D-7})$$

As the client data has to fit into the payload area of the server signal, the maximum value of C_n and as such the maximum client bit rate is limited by the size of the server payload area.

$$C_n(t) \leq P_{server} \quad (\text{D-8})$$

$$f_{client} \leq f_{server} \times \frac{P_{server}}{B_{server}} \times n \quad (\text{D-9})$$

P_{server} : maximum number of (n bits) data entities in the server payload area

The client and server bit rate are independent. This allows specifying the server bit rate independently from the client bit rates. Furthermore, client clock impairments are not seen at the server clock.

If the client or server bit rate changes due to client or server frequency tolerances, c_n and $C_n(t)$ change accordingly. A special procedure has to take care that $C_n(t)$ is changed fast enough to the correct value during start-up or during a step in the client bit rate (e.g., when the client signal is replaced by its AIS signal or the AIS signal is replaced by the client signal). This procedure may be designed to prevent buffer over-/underflow, or an additional buffer over-/underflow prevention method has to be deployed.

A transparent mapping has to determine $C_n(t)$ on a server (multi)frame per (multi)frame base.

In order to extract the correct number of client information entities at the de-mapper, $C_n(t)$ has to be transported in the overhead area of the server frame or multiframe from the mapper to the de-mapper.

Figure D.1 shows the generic functionality of the mapper and de-mapper circuit.

At the mapper, $C_n(t)$ is determined based on the client and server clocks. The client data is constantly written into the buffer memory. The read out is controlled by the value of $C_n(t)$.

At the de-mapper, $C_n(t)$ is extracted from the overhead. $C_n(t)$ controls the write enable signal for the buffer. The client clock is generated based on the server clock and the value of $C_n(t)$.

$C_n(t)$ has to be determined first, then it has to be inserted into the overhead and afterwards $C_n(t)$ client data entities have to be inserted into the payload area of the server as shown in Figure D.2.

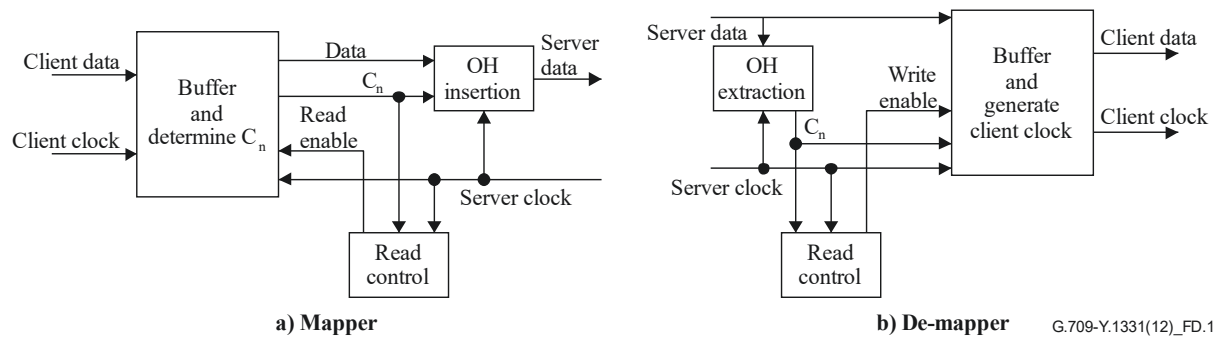


Figure D.1 – Generic functionality of a mapper/de-mapper circuit

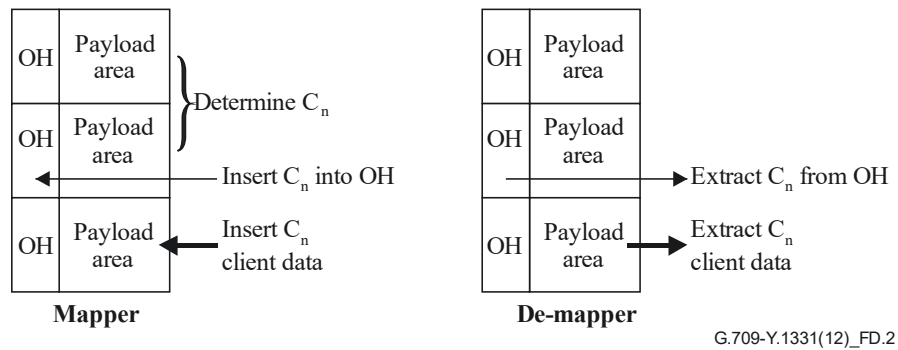


Figure D.2 – Processing flow

$C_n(t)$ client data entities are mapped into the payload area of the server frame or multiframe using a sigma-delta data/stuff mapping distribution. It provides a distributed mapping as shown in Figure D.3. Payload field j ($j = 1 \dots P_{\text{server}}$) carries:

$$- \text{client data (D)} \quad \text{if } (j \times C_n(t)) \bmod P_{\text{server}} < C_n(t) \quad (\text{D-10})$$

$$- \text{stuff (S)} \quad \text{if } (j \times C_n(t)) \bmod P_{\text{server}} \geq C_n(t). \quad (\text{D-11})$$

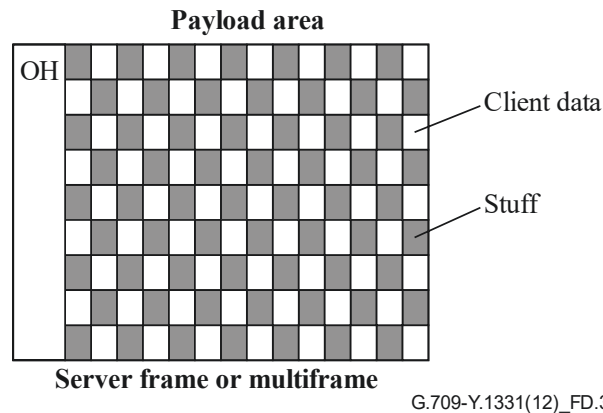


Figure D.3 – Sigma-delta based mapping

$C_n(t)$ client data entities have to be distributed over P_{server} locations. A client data entity has therefore to be inserted with a spacing of $\frac{P_{\text{server}}}{C_n(t)}$. This is normally not an integer value, however it can be emulated by an integer calculation using the sigma-delta method based on an overflow accumulator as shown in Figure D.4.

The accumulator memory is reset to 0 at every frame start of the server frame. At every location of the payload area, $C_n(t)$ is added to the memory and the result is compared with P_{server} . If the result is lower than P_{server} , it is stored back into the memory and no client data is indicated for this payload position. If it is equal or greater than P_{server} , P_{server} is subtracted from the result and the new result is stored back in the memory. In addition, client data is indicated for the client position.

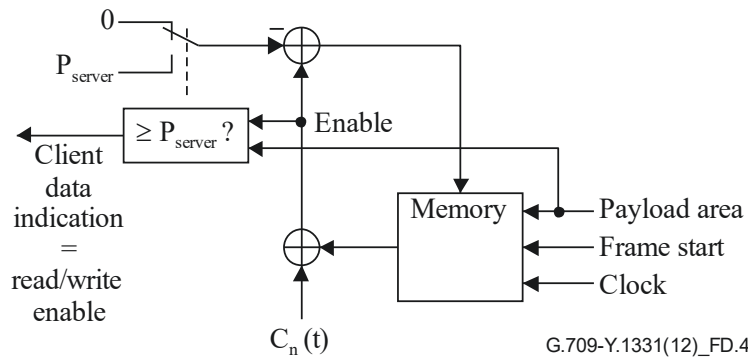


Figure D.4 – Sigma-delta accumulator

As the same start value and $C_n(t)$ are used at the mapper and de-mapper the same results are obtained and interworking is achieved.

D.2 Applying GMP in OTN

Clauses 17.7, 19.6 and 20.5 specify GMP as the asynchronous generic mapping method for the mapping of CBR client signals into OPUk, the mapping of ODUk signals into a server OPUk (via the ODTUk.ts) and the mapping of ODUk signals into an OPUCn (via ODTUCn.ts).

NOTE – GMP complements the traditional asynchronous client/server specific mapping method specified in clauses 17.6 and 19.5. GMP is intended to provide the justification of new CBR type client signals into OPUk.

Asynchronous mappings in the OTN have a default 8-bit timing granularity. Such 8-bit timing granularity is supported in GMP by means of a c_n with $n=8$ (c_8). The jitter/wander requirements for some of the OTN client signals are such that for those signals an 8-bit timing granularity may not be sufficient. For such a case, a 1-bit timing granularity is supported in GMP by means of c_n with $n=1$ (c_1).

M-byte granularity mapping

Clauses 17.7 and 19.6 specify that the mapping of CBR client bits into the payload of an OPUk and the mapping of ODUj bits into the payload of an ODTUk.ts is performed with $8 \times M$ -bit (M-byte) granularity.

The insertion of CBR client data into the payload area of the OPUk frame and the insertion of ODUj data into the payload area of the ODTUk.ts multiframe at the mapper is performed in M-byte (or m-bit, $m = 8 \times M$) data entities, denoted as $C_m(t)$. The remaining $C_{nD}(t)$ data entities are signalled in the justification overhead as additional timing/phase information.

$$c_m = \left(\frac{n \times c_n}{m} \right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{m} \right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{8 \times M} \right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M} \right) \quad (D-12)$$

As only an integer number of m-bit data entities can be transported per server frame or multiframe, the integer value $C_m(t)$ of c_m has to be used. Since it is required that no information is lost, the rounding process to the integer value has to take care of the truncated part, e.g., a c_m with a value of 10.25 has to be represented by the integer sequence 10,10,10,11.

$$C_m(t) = \text{int}(c_m) = \text{int} \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M} \right) \quad (D-13)$$

For the case c_m is not an integer, $C_m(t)$ will vary between:

$$C_m(t) = \text{floor}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \text{ and } C_m(t) = \text{ceiling}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/8}{M}\right) \quad (\text{D-14})$$

The remainder of c_n and $C_m(t)$ is:

$$c_{nD} = c_n - \left(\frac{8 \times M}{n} \times C_m(t)\right) \quad (\text{D-15})$$

As only an integer number of c_{nD} n-bit data entities can be signalled per server frame or multiframe, the integer value $C_{nD}(t)$ of c_{nD} has to be used.

$$C_{nD}(t) = \text{int}(c_n) - \left(\frac{8 \times M}{n} \times C_m(t)\right) = C_n(t) - \left(\frac{8 \times M}{n} \times C_m(t)\right) \quad (\text{D-16})$$

$C_{nD}(t)$ is a number between $1 - \frac{8 \times M}{n}$ and $\frac{8 \times M}{n} - 1$.

16M-byte granularity mapping

Clause 20.5 specifies that the mapping of ODUk bits into the payload of an ODUCn.ts is performed with $128 \times M$ -bit (16M-byte) granularity.

The insertion of ODUk data into the payload area of the ODTUCn.ts multiframe at the mapper is performed in 16M-byte (or m-bit, $m = 128 \times M$) data entities, denoted as $c_m(t)$. The remaining $C_{nD}(t)$ data entities are signalled in the justification overhead as additional timing/phase information.

$$c_m = \left(\frac{n \times c_n}{m}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{m}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}}{128 \times M}\right) = \left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/128}{M}\right) \quad (\text{D-20})$$

As only an integer number of m-bit data entities can be transported per server frame or multiframe, the integer value $C_m(t)$ of c_m has to be used. Since it is required that no information is lost, the rounding process to the integer value has to take care of the truncated part, e.g., a c_m with a value of 10.25 has to be represented by the integer sequence 10,10,10,11.

$$C_m(t) = \text{int}(c_m) = \text{int}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/128}{M}\right) \quad (\text{D-21})$$

For the case c_m is not an integer, $C_m(t)$ will vary between:

$$C_m(t) = \text{floor}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/128}{M}\right) \text{ and } C_m(t) = \text{ceiling}\left(\frac{f_{client}}{f_{server}} \times \frac{B_{server}/128}{M}\right) \quad (\text{D-22})$$

The remainder of c_n and $C_m(t)$ is:

$$c_{nD} = c_n - \left(\frac{128 \times M}{n} \times C_m(t)\right) \quad (\text{D-23})$$

As only an integer number of c_{nD} n-bit data entities can be signalled per server frame or multiframe, the integer value $C_{nD}(t)$ of c_{nD} has to be used.

$$C_{nD}(t) = \text{int}(c_n) - \left(\frac{128 \times M}{n} \times C_m(t)\right) = C_n(t) - \left(\frac{128 \times M}{n} \times C_m(t)\right) \quad (\text{D-24})$$

$C_{nD}(t)$ is a number between $1 - \frac{128 \times M}{n}$ and $\frac{128 \times M}{n} - 1$.

As the client data has to fit into the payload area of the server signal, the maximum value of C_m and as such the maximum client bit rate is limited by the size of the server payload area.

$$C_m(t) \leq P_{m,server} \quad (D-17)$$

$P_{m,server}$: maximum number of (m bits) data entities in the server payload area

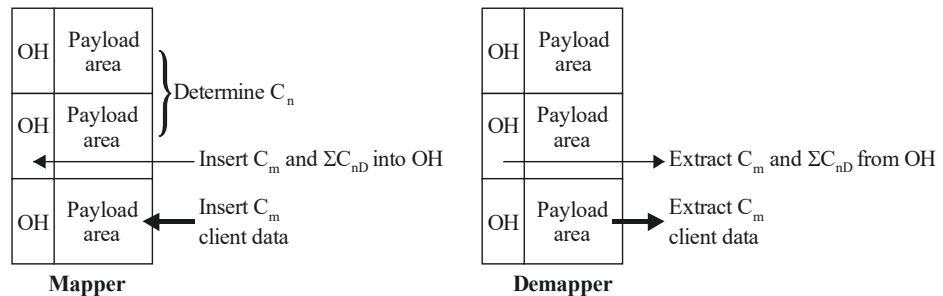
In order to extract the correct number of client information entities at the de-mapper, $C_m(t)$ has to be transported in the overhead area of the server frame or multiframe from the mapper to the de-mapper.

At the mapper, $C_n(t)$ is determined based on the client and server clocks. The client data is constantly written into the buffer memory. The read out is controlled by the value of $C_m(t)$.

At the de-mapper, $C_m(t)$ and $C_{nD}(t)$ are extracted from the overhead and used to compute $C_n(t)$. $C_m(t)$ controls the write enable signal for the buffer. The client clock is generated based on the server clock and the value of $C_n(t)$.

$C_n(t)$ has to be determined first, then it has to be inserted into the overhead as $C_m(t)$ and $\Sigma C_{nD}(t)$ and afterwards $C_m(t)$ client data entities have to be inserted into the payload area of the server as shown in Figure D.5.

The $C_n(t)$ value determines the $C_m(t)$ and $C_{nD}(t)$ values; $C_m(t) = \text{floor}(n/m \times C_n(t))$ and $C_{nD}(t) = C_n(t) - (m/n \times C_m(t))$. The values of $C_{nD}(t)$ are accumulated and if $\Sigma C_{nD}(t) \geq m/n$ then m/n is subtracted from $\Sigma C_{nD}(t)$ and $C_m(t)$ is incremented with +1. These latter two values are then encoded in the overhead bytes. This $C_m(t)$ value is applied as input to the sigma-delta process.



G.709-Y.1331(12)_FD.5

Figure D.5 – Processing flow for GMP in OTN

During start-up or during a step in the client bit rate, the value of $C_n(t)$ will not match the actual number of n-bit client data entities arriving at the mapper buffer and the $C_n(t)$ determination process has to adjust its value to the actual number of n-bit client data entities arriving. This adjustment method is implementation specific. During the mismatch period, the mapper buffer fill level may increase if more n-bit client data entities arrive per multiframe than there are transmitted, or decrease if less n-bit client data entities arrive per multiframe than there are transmitted.

To prevent overflow or underflow of the mapper buffer and thus data loss, the fill level of the mapper buffer has to be monitored. For the case where too many m-bit client data entities are in the buffer, it is necessary to insert temporarily more m-bit client data entities in the server (multi)frame(s) than required by $C_n(t)$. For the case too few m-bit client data entities are in the buffer, it is necessary to insert temporarily fewer m-bit client data entities in the server (multi)frame(s) than required by $C_n(t)$. This behaviour is similar to the behaviour of AMP under these conditions.

The OTN supports a number of client signal types for which transfer delay (latency) and transfer delay variation are critical parameters. Those client signal types require that the transfer delay introduced by the mapper plus de-mapper buffers is minimized and that the delay variation introduced by the mapper plus de-mapper buffers is minimized.

In steady state periods, $C_n(t)$ is a value in the range $C_{n,\min}$ to $C_{n,\max}$. A value outside this range indicates that there is a misalignment of the expected client bit rate and the actual client bit rate. During transient periods after e.g., a frequency step, $C_n(t)$ may be temporarily outside the range $C_{n,\min}$ to $C_{n,\max}$.

$C_m(t)$ client data entities are mapped into the payload area of the server frame or multiframe using a sigma-delta data/stuff mapping distribution. It provides a distributed mapping as shown in Figure D.3. Payload field j ($j = 1 \dots P_{m,\text{server}}$) carries

$$\text{client data (D)} \quad \text{if } (j \times C_m(t)) \bmod P_{m,\text{server}} < C_m(t); \quad (\text{D-18})$$

$$\text{stuff (S)} \quad \text{if } (j \times C_m(t)) \bmod P_{m,\text{server}} \geq C_m(t). \quad (\text{D-19})$$

Values of n , m , M , f_{client} , f_{server} , T_{server} , B_{server} , and $P_{m,\text{server}}$ for OPUk and ODTUk.ts

The values for n , m , M , f_{client} , f_{server} , T_{server} , B_{server} , and $P_{m,\text{server}}$ are specified in Table D.1.

Table D.1 – OPUk, ODTUk.ts and ODTUCn.ts GMP parameter values

GMP parameter	CBR client into OPUk	ODUj into OPUk (ODTUk.ts)	ODUk into OPUCn (ODTUCn.ts)
n	8 (default) 1 (client specific)	8	8
m	$m = 8 \times M$ OPU0: $8 \times 1 = 8$ OPU1: $8 \times 2 = 16$ OPU2: $8 \times 8 = 64$ OPU3: $8 \times 32 = 256$ OPU4: $8 \times 80 = 640$	$m = 8 \times M$ ODTU2.ts: $8 \times \text{ts}$ ODTU3.ts: $8 \times \text{ts}$ ODTU4.ts: $8 \times \text{ts}$	$m = 128 \times M$ ODTUCn.ts: $128 \times \text{ts}$
f_{client}	CBR client bit rate and tolerance	ODUj bit rate and tolerance (Table 7-2)	ODUk bit rate and tolerance (Table 7-2)
f_{server}	OPUk Payload bit rate and tolerance (Table 7-3)	ODTUk.ts Payload bit rate and tolerance (Table 7-7)	ODTUCn.ts Payload bit rate and tolerance (Table 7-7)
T_{server}	ODUk/OPUk frame period (Table 7-4)	OPUk multiframe period (Table 7-6)	OPUCn multiframe period (Table 7-6)
B_{server}	OPU0: 8×15232 OPU1: 8×15232 OPU2: 8×15232 OPU3: 8×15232 OPU4: 8×15200	ODTU2.ts: $8 \times \text{ts} \times 15232$ ODTU3.ts: $8 \times \text{ts} \times 15232$ ODTU4.ts: $8 \times \text{ts} \times 15200$	ODTUCn.ts: $128 \times \text{ts} \times 952$
$P_{m,\text{server}}$	OPU0: 15232 OPU1: 7616 OPU2: 1904 OPU3: 476 OPU4: 190	ODTU2.ts: 15232 ODTU3.ts: 15232 ODTU4.ts: 15200	ODTUCn.ts: 952
ΣC_{8D} range	OPU0: N/A OPU1: 0 to +1 OPU2: 0 to +7 OPU3: 0 to +31 OPU4: 0 to +79	ODTUk.1: N/A ODTUk.2: 0 to +1 ODTUk.3: 0 to +2 ODTUk.4: 0 to +3 : ODTUk.8: 0 to +7 : ODTUk.32: 0 to +31 :	ODTUCn.1: 0 to +15 ODTUCn.2: 0 to +31 ODTUCn.3: 0 to +47 ODTUCn.4: 0 to +59 : ODTUCn.20n-1: 0 to +320n-1 ODTUCn.20n: 0 to +320n

Table D.1 – OPUk, ODTUk.ts and ODTUCn.ts GMP parameter values

GMP parameter	CBR client into OPUk	ODUj into OPUk (ODTUk.ts)	ODUk into OPUCn (ODTUCn.ts)
		ODTUk.79: 0 to +78 ODTUk.80: 0 to +79	
ΣC_{1D} range (for selected clients)	OPU0: 0 to +7 OPU1: 0 to +15 OPU2: 0 to +63 OPU3: 0 to +255 OPU4: 0 to +639	Not applicable	Not applicable

D.3 $C_m(t)$ encoding and decoding

$C_m(t)$ is encoded in the ODTUk.ts justification control bytes JC1, JC2 and JC3 specified in clause 19.4 for the 14-bit count and clause 20.4 for the 10-bit count field.

$C_m(t)$ is an L-bit binary count of the number of groups of m OPU payload bits that carry m client bits; it has values between $\text{Floor}(C_{m,\min})$ and $\text{Ceiling}(C_{m,\max})$, which are client specific. The C_i ($i=1..L$) bits that comprise $C_m(t)$ are used to indicate whether the $C_m(t)$ value is incremented or decremented from the value in the previous frame, that is indicated by $C_m(t-1)$. Tables D.2 and D.3 show the inversion patterns for the C_i bits of $C_m(t-1)$ that are inverted to indicate an increment or decrement of the $C_m(t)$ value. Table D.2 shows the inversion patterns for the 14-bit count and Table D.3 shows the inversion patterns for the 10-bit count. An "I" entry in the table indicates an inversion of that bit.

The bit inversion patterns apply to the $C_m(t-1)$ value, prior to the increment or decrement operation that is signalled by the inversion pattern when $|C_m(t) - C_m(t-1)| \leq 2$ (except $C_m(t) - C_m(t-1) = 0$). The incremented or decremented $C_m(t)$ value becomes the base value for the next GMP overhead transmission.

- When $0 < C_m(t) - C_m(t-1) \leq 2$, indicating an increment of +1 or +2, a subset of the C_i bits containing $C_m(t-1)$ is inverted as specified in Table D.2 or Table D.3 and the increment indicator (II) bit is set to 1.
- When $0 > C_m(t) - C_m(t-1) \geq -2$, indicating a decrement of -1 or -2, a subset of C_i bits containing $C_m(t-1)$ is inverted as specified in Table D.2 or Table D.3 and the decrement indicator (DI) bit is set to 1.
- When the value of $C_m(t)$ is changed with a value larger than +2 or –2 from the value of $C_m(t-1)$, both the II and DI bits are set to 1 and the C_i bits contain the new $C_m(t)$ value. The associated CRC in JC3 verifies whether the $C_m(t)$ value has been received correctly.
- When the value of $C_m(t)$ is unchanged from the value of $C_m(t-1)$, both the II and DI bits are set to 0.

The above encoding process is illustrated in Figure D.6.

Table D.2 – 14-bit $C_m(t)$ increment and decrement indicator patterns

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Change
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	I	U	I	U	1	0	+1
U	I	U	I	U	I	U	I	U	I	U	I	U	I	0	1	-1
U	I	I	U	U	I	I	U	U	I	I	U	U	I	1	0	+2
I	U	U	I	I	U	U	I	I	U	U	I	I	U	0	1	-2
binary value														1	1	More than +2/-2
NOTE																
– I indicates inverted C_i bit																
– U indicates unchanged C_i bit																

The CRC-8 located in JC3 is calculated over the JC1 and JC2 bits. The CRC-8 uses the $g(x) = x^8 + x^3 + x^2 + 1$ generator polynomial, and is calculated as follows:

- 1) The JC1 and JC2 octets are taken in network octet order, most significant bit first, to form a 16-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 15.
- 2) $M(x)$ is multiplied by x^8 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 7 or less.
- 3) The coefficients of $R(x)$ are considered to be an 8-bit sequence, where x^7 is the most significant bit.
- 4) This 8-bit sequence is the CRC-8 where the first bit of the CRC-8 to be transmitted is the coefficient of x^7 and the last bit transmitted is the coefficient of x^0 .

The de-mapper process performs steps 1-3 in the same manner as the mapper process, except that here, the $M(x)$ polynomial of step 1 includes the CRC bits of JC3, resulting in $M(x)$ having degree 23. In the absence of bit errors, the remainder shall be 0000 0000.

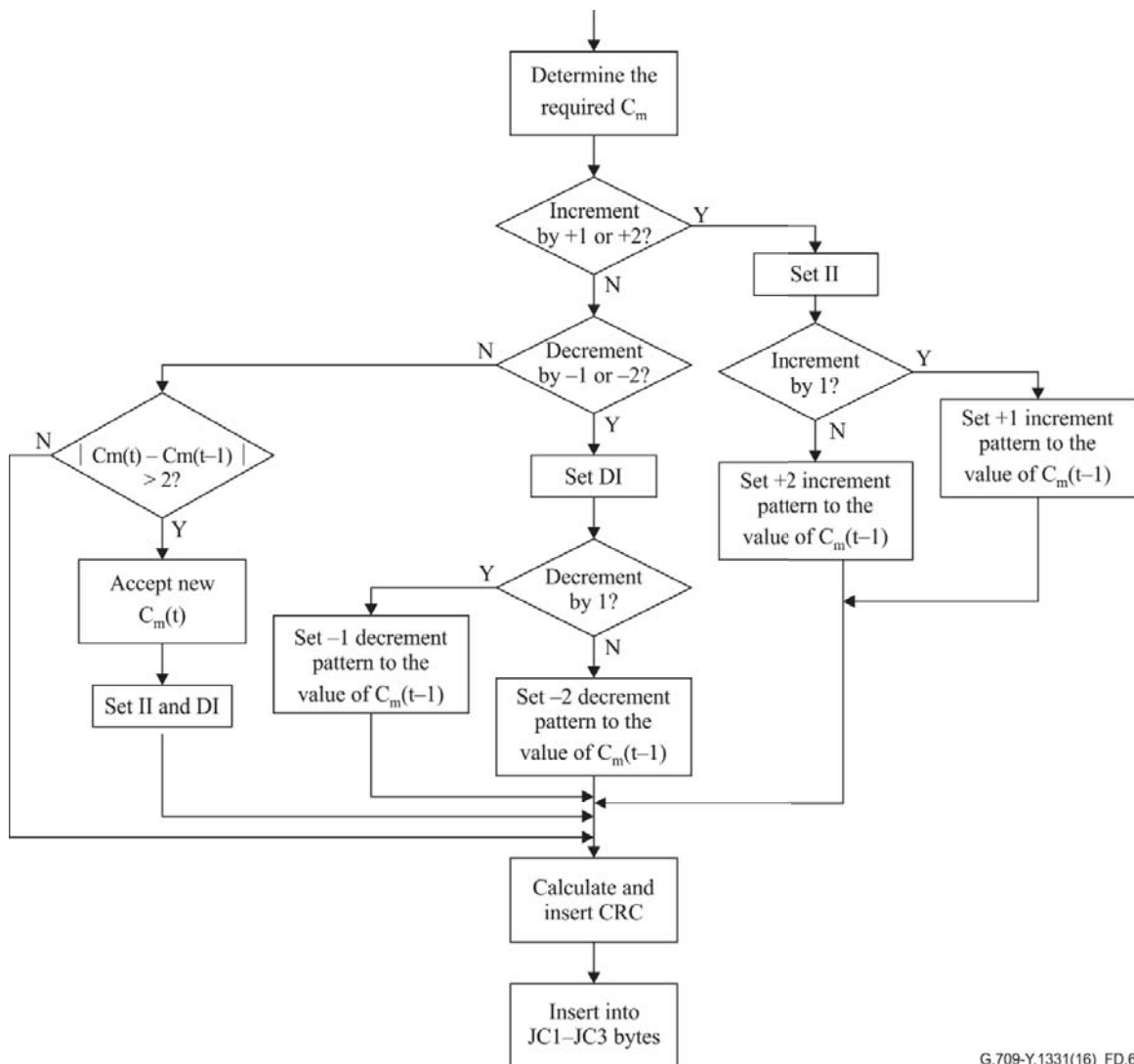
Table D.3 – 10-bit $C_m(t)$ increment and decrement indicator patterns

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	II	DI	Change
U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	1	0	+1
I	U	U	I	U	I	I	U	U	I	0	1	-1
U	I	U	I	I	U	U	I	U	I	1	0	+2
U	I	I	U	U	I	U	I	I	U	0	1	-2
binary value										1	1	More than +2/-2
NOTE												
– I indicates inverted C_i bit												
– U indicates unchanged C_i bit												

The CRC-6 located in JC3 is calculated over bits 3-8 of JC1 and JC2 (i.e., the bits containing the GMP overhead value shown in Table D.3). The CRC-6 uses the $g(x) = x^6 + x^3 + x^2 + 1$ generator polynomial, and is calculated as follows:

- 1) The JC1 and JC2 octets are taken in network octet order, most significant bit first, such that bit 3 of JC1 through bit 8 of JC2 form a 12-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 11.
- 2) $M(x)$ is multiplied by x^6 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 5 or less.
- 3) The coefficients of $R(x)$ are considered to be a 6-bit sequence, where x^5 is the most significant bit.
- 4) This 6-bit sequence is the CRC-6 where the first bit of the CRC-6 to be transmitted is the coefficient of x^5 and the last bit transmitted is the coefficient of x^0 .

The de-mapper process performs steps 1-3 in the same manner as the mapper process, except that here, the $M(x)$ polynomial of step 1 includes the CRC bits of JC3, resulting in $M(x)$ having degree 17. In the absence of bit errors, the remainder shall be 000000.



G.709-Y.1331(16)_FD.6

Figure D.6 – JC1, JC2 and JC3 generation

A parallel logic implementation of the source CRC-8 and CRC-6 are illustrated in Appendix VI.

The GMP sink synchronizes its $C_m(t)$ value to the GMP source through the following process, which is illustrated in Figure D.7 and the equivalent process in Figure D.8.

When the received JC octets contain $II = DI$ and a valid CRC-8, the GMP sink accepts the received C1-CL as its $C_m(t)$ value for the next frame. At this point the GMP sink is synchronized to the GMP source. When $II \neq DI$ with a valid CRC in the current received frame (frame i), the GMP sink must examine the received JC octets in the next frame (frame $i+1$) in order to obtain $C_m(t)$ synchronization. $II \neq DI$ in frame i indicates that the source is performing a count increment or decrement operation that will modify the $C_m(t)$ value it sends in frame $i+1$. Since this modification to the $C_m(t)$ will affect the count LSBs, the GMP sink uses the LSBs, II, and DI in frame i to determine its synchronization hunt state when it receives frame $i+1$. Specifically, in Figure D.7, the Hunt state (A-F) is determined using C13, C14, II and DI for the 14-bit count and C9, C10, II and DI for the 10-bit count. Equivalently, in Figure D.8, the Hunt state (A or B) is determined using C14, II, and DI for the 14-bit count and C10, II, and DI for the 10-bit count. If $II = DI$ with a valid CRC in frame $i+1$, $C_m(t)$ synchronization is achieved by directly accepting the received C1-CL as the new $C_m(t)$. If $II \neq DI$ with a valid CRC in frame $i+1$, the sink uses the new LSBs, II and DI values to determine whether the source is communicating an increment or decrement operation and the magnitude of the increment/decrement step. This corresponds to the transition to the lower row of states (the "S" states) in Figure D.7 and Figure D.8. At this point, the GMP sink has identified the type of increment or decrement operation that is being signalled in frame $i+1$. As shown in Figure D.9, the sink then applies the appropriate bit inversion pattern from Table D.2 to the received C1-C14 field or the bit inversion pattern from Table D.3 to the received C1-C10 field to determine the transmitted $C_m(t)$ value. Synchronization has now been achieved since the GMP sink has determined the current $C_m(t)$ and knows the expected $C_m(t)$ change in frame $i+2$.

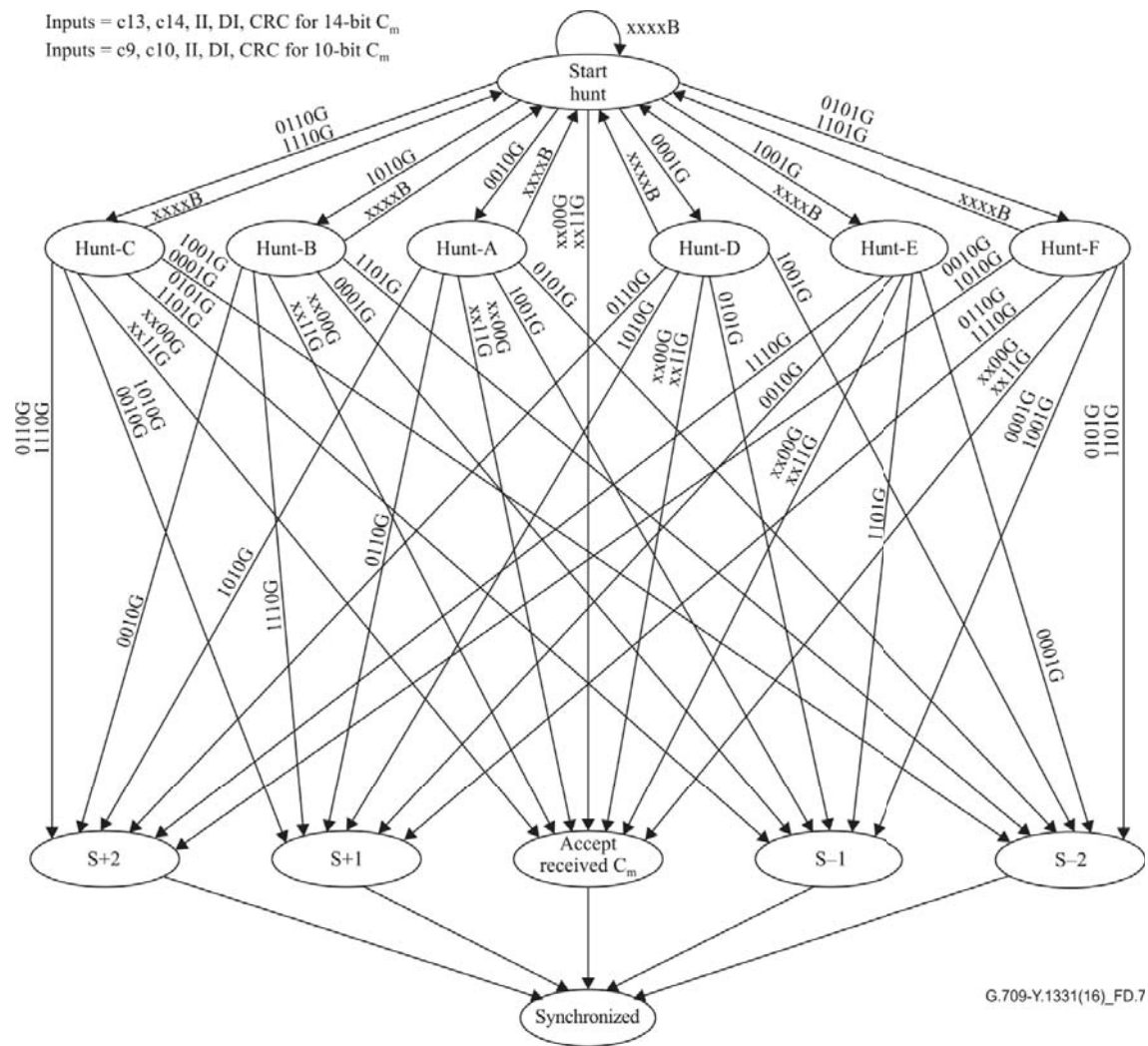


Figure D.7 – GMP sink count synchronization process representation using the two least significant bits of the count as inputs

Optimization has shown that Figure D.8 provides the same function. Either one can be chosen. For historical reasons both are kept.

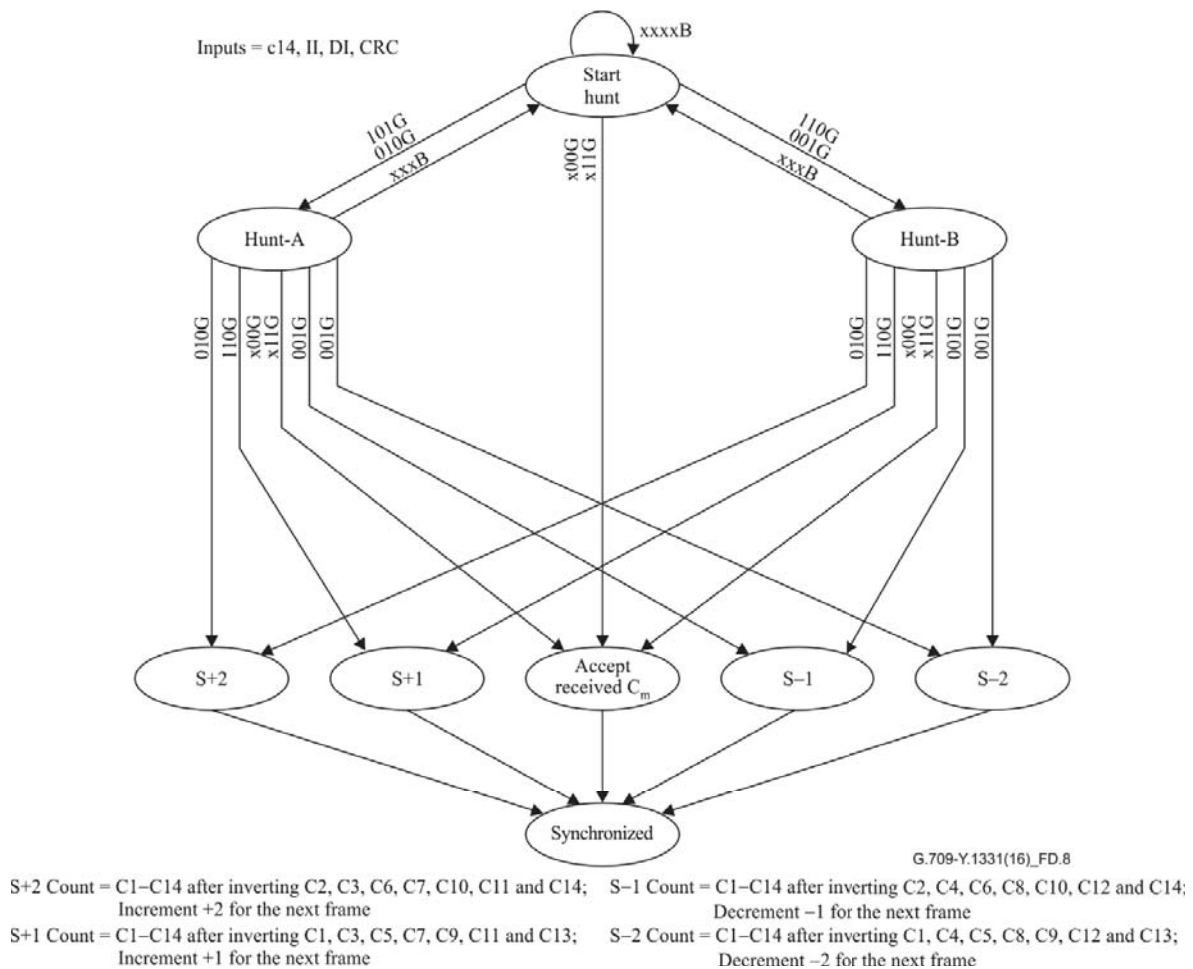


Figure D.8 – GMP sink count synchronization process diagram

14-bit count		10-bit count	
S state	Interpretation / Action	S state	Interpretation / Action
S+2	Count = C1–C14 after inverting C2, C3, C6, C7, C10, C11, & C14; Increment +2 for the next frame	S+2	Count = C1–C10 after inverting C2, C4, C5, C8 & C10; Increment +2 for the next frame
S+1	Count = C1–C14 after inverting C1, C3, C5, C7, C9, C11, & C13; Increment +1 for the next frame	S+1	Count = C1–C10 after inverting C1, C3, C5, C7 & C9; Increment +1 for the next frame
S–1	Count = C1–C14 after inverting C2, C4, C6, C8, C10, C12, & C14; Decrement –1 for next frame	S–1	Count = C1–C10 after inverting C1, C4, C6, C7 & C10; Decrement –1 for next frame
S–2	Count = C1–C14 after inverting C1, C4, C5, C8, C9, C12, C13; Decrement –2 for next frame	S–2	Count = C1–C10 after inverting C2, C3, C6, C8 & C9; Decrement –2 for next frame

Figure D.9 – "S"-state interpretation for Figures D.7 and D.8

Note that the state machine of Figure D.7 or Figure D.8 can also be used for off-line synchronization checking.

When the GMP sink has synchronized its $C_m(t)$ value to the GMP source, it interprets the received JC octets according to the following principles:

- When the CRC is good and $II = DI$, the GMP sink accepts the received $C_m(t)$ value.
- When the CRC is good and $II \neq DI$, the GMP sink compares the received $C_m(t)$ value to its expected $C_m(t)$ value to determine the difference between these values. This difference is compared to the bit inversion patterns of Table D.2 or Table D.3 to determine the increment or decrement operation sent by the source and updates its $C_m(t)$ accordingly. Since the CRC is good, the sink can use either JC1 or JC2 for this comparison.
- When the CRC is bad, the GMP sink compares the received $C_m(t)$ value to its expected $C_m(t)$ value. The sink then compares the difference between these values, per Table D.2 or Table D.3, to the valid bit inversion patterns in JC1, and the bit inversion, II and DI pattern in JC2.
 - If JC1 contains a valid pattern and JC2 does not, the sink accepts the corresponding increment or decrement indication from JC1 and updates its $C_m(t)$ accordingly.
 - If JC2 contains a valid pattern and JC1 does not, the sink accepts the corresponding increment or decrement indication from JC2 and updates its $C_m(t)$ accordingly.
 - If both JC1 and JC2 contain valid patterns indicating the same increment or decrement operation, this indication is accepted and the sink updates its $C_m(t)$ accordingly.
 - If neither JC1 nor JC2 contain valid patterns, the sink shall keep its current count value and begin the search for synchronization.

NOTE – If JC1 and JC2 each contain valid patterns that are different from each other, the receiver can either keep the current $C_m(t)$ value and begin a synchronization search, or it can use the CRC to determine whether JC1 or JC2 contains the correct pattern.

The GMP sink uses the updated $C_m(t)$ value to extract the client data from the next OPU frame or ODTUk.ts multiframe.

D.4 $\Sigma C_{nd}(t)$ encoding and decoding

D.4.1 $\Sigma C_{nd}(t)$ encoding and decoding for OPUk

The cumulative value of $C_{nd}(t)$ ($\Sigma C_{nd}(t)$) is encoded in bits 4-8 of the OPUk and ODTUk.ts justification control bytes JC4, JC5 and JC6. Bits D1 to D10 in JC4 and JC5 carry the value of $\Sigma C_{nd}(t)$. Bit D1 carries the most significant bit and bit D10 carries the least significant bit.

The CRC-5 located in JC6 is calculated over the D1-D10 bits in JC4 and JC5. The CRC-5 uses the $g(x) = x^5 + x + 1$ generator polynomial, and is calculated as follows:

- 1) The JC4 bits 4-8 and JC5 bits 4-8 octets are taken in network transmission order, most significant bit first, to form a 10-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 9.
- 2) $M(x)$ is multiplied by x^5 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 4 or less.
- 3) The coefficients of $R(x)$ are considered to be a 5-bit sequence, where x^4 is the most significant bit.
- 4) This 5-bit sequence is the CRC-5 where the first bit of the CRC-5 to be transmitted is the coefficient of x^4 and the last bit transmitted is the coefficient of x^0 .

The de-mapper process performs steps 1-3 in the same manner as the mapper process, except that here, the $M(x)$ polynomial of step 1 includes the CRC bits of JC6, resulting in $M(x)$ having degree 14. In the absence of bit errors, the remainder shall be 00000.

A parallel logic implementation of the source CRC-5 is illustrated in Appendix VI.

D.4.2 $\Sigma C_{nD}(t)$ encoding and decoding for OPU C_n

The cumulative value of $C_{nD}(t)$ ($\Sigma C_{nD}(t)$) is encoded in the ODTUC n .ts justification control bytes JC1, JC2, JC3, JC4, JC5 and JC6. Bits D1 to D18 carry the value of $\Sigma C_{nD}(t)$. Bit D1 carries the most significant bit and bit D10 carries the least significant bit. As shown in Figure 20-7, bits D1 to D7 are located in bits 2-8 of JC4, bits D8 to D9 are located in bits 1-2 of JC1, bits D10 to D16 are located in JC5, and bits D17 to D18 are located in bits 1-2 of JC2.

The CRC-9 located in bits 2-8 of JC6 and bits 1-2 of JC3 is calculated over the D1-D18 bits in JC4, JC1, JC5 and JC2. The CRC-9 uses the $g(x) = x^9 + x^3 + x^2 + 1$ generator polynomial, and is calculated as follows:

- 1) The JC4 bits 2-8, JC1 bits 1-2, JC5 bits 2-8, and JC2 bits 1-2 are taken in network transmission order, most significant bit first, to form an 18-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 17.
- 2) $M(x)$ is multiplied by x^9 and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 8 or less.
- 3) The coefficients of $R(x)$ are considered to be a 9-bit sequence, where x^8 is the most significant bit.
- 4) This 9-bit sequence is the CRC-9 where the first bit of the CRC-9 to be transmitted is the coefficient of x^8 and the last bit transmitted is the coefficient of x^0 .

The de-mapper process performs steps 1-3 in the same manner as the mapper process, except that here, the $M(x)$ polynomial of step 1 includes the CRC bits of JC6 and JC3, resulting in $M(x)$ having degree 26. In the absence of bit errors, the remainder shall be 000000000.

A parallel logic implementation of the source CRC-9 is illustrated in Appendix VI.

Annex E

Adaptation of parallel 64B/66B encoded clients

(This annex forms an integral part of this Recommendation.)

E.1 Introduction

IEEE 40GBASE-R and 100GBASE-R interfaces specified in [IEEE 802.3] are parallel interfaces intended for short-reach (up to 40 km) interconnection of Ethernet equipment. This annex describes the process of converting the parallel format of these interfaces into a serial bit stream to be carried over the OTN.

The order of transmission of information in all the diagrams in this annex is first from left to right and then from top to bottom.

E.2 Clients signal format

40GBASE-R and 100GBASE-R clients are initially parallel interfaces, but in the future they may be serial interfaces. Independent of whether these interfaces are parallel or serial, or what the parallel interface lane count is, 40GBASE-R signals are comprised of four PCS lanes, and 100GBASE-R signals are comprised of twenty PCS lanes. If the number of physical lanes on the interface is fewer than the number of PCS lanes, the appropriate number of PCS lanes is bit-multiplexed onto each physical lane of the interface. Each PCS lane consists of 64B/66B encoded data with a PCS lane alignment marker inserted on each lane once per 16384 66-bit blocks. The PCS lane alignment marker itself is a special format 66B codeword.

The use of this adaptation for 40GBASE-R into OPU3 also applies the transcoding method that appears in Annex B and the framing method of Annex F. The adaptation described in this annex alone can be used for the adaptation of 100GBASE-R into OPU4.

E.3 Client frame recovery

Client framing recovery consists of the following:

- bit-disinterleave the PCS lanes, if necessary. This is necessary whenever the number of PCS lanes and the number of physical lanes is not equal, and is not necessary when they are equal (e.g., a 4-lane 40GBASE-R interface);
- recover 64B/66B block lock as per the state diagram in Figure 82-10 of [IEEE 802.3];
- recover lane alignment marker framing on each PCS lane as per the state diagram in Figure 82-11 of [IEEE 802.3];
- reorder and de-skew the PCS lanes into a serialized stream of 66B blocks (including lane alignment markers). Figure E.1 illustrates the ordering of 66B blocks after the completion of this process for an interface with p PCS lanes.

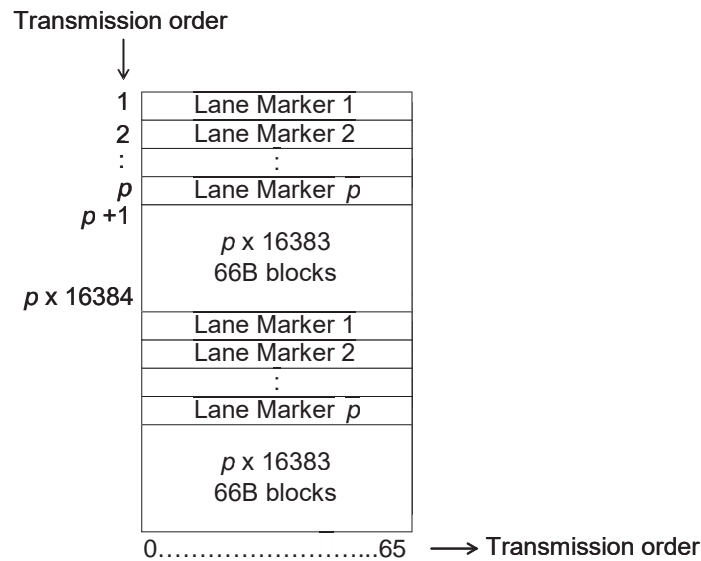


Figure E.1 – De-skewed/serialized stream of 66B blocks

Each 66B codeword is one of the following:

- a set of eight data bytes with a sync header of "01";
- a control block (possibly including seven or fewer data octets) beginning with a sync header of "10";
- a PCS lane alignment marker, also encoded with a sync header of "10". Of the 8 octets following the sync header, 6 octets have fixed values allowing the PCS lane alignment markers to be recognized (see Tables E.1 and E.2). The fourth octet following the sync header is a BIP-8 calculated over the data from one alignment marker to the next as defined in Table 82-4 of [IEEE 802.3]. The eighth octet is the complement of this BIP-8 value to maintain DC balance. Note that the intended operation is to pass these BIP-8 values transparently as they are used for monitoring the error ratio of the Ethernet link between Ethernet PCS sublayers. For the case of 100GBASE-R, the BIP-8 values are not manipulated by the mapping or de-mapping procedure. For the case of 40GBASE-R, a BIP-8 compensation is done as described in clause E.4.1.

For all-data blocks and control blocks, the 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane alignment markers) according to the polynomial $G(x) = 1 + x^{39} + x^{58}$.

After 64B/66B block lock recovery as per the state diagram in Figure 82-10 of [IEEE 802.3] to the single-lane received aggregate signal, these 66B blocks are re-distributed to PCS lanes at the egress interface. The 66B blocks (including PCS lane alignment markers) resulting from the decoding process are distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

E.3.1 40GBASE-R client frame recovery

PCS lane alignment markers have the values shown in Table E.1 for 40GBASE-R signals which use PCS lane numbers 0-3.

Table E.1 – PCS lane alignment marker format for 40GBASE-R

Lane Number	SH	Encoding $\{M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7\}$
0	10	0x90, 0x76, 0x47, BIP ₃ , 0x6f, 0x89, 0xb8, BIP ₇
1	10	0xf0, 0xc4, 0xe6, BIP ₃ , 0x0f, 0x3b, 0x19, BIP ₇
2	10	0xc5, 0x65, 0x9b, BIP ₃ , 0x3a, 0x9a, 0x64, BIP ₇
3	10	0xa2, 0x79, 0x3d, BIP ₃ , 0x5d, 0x86, 0xc2, BIP ₇

Since a 40GBASE-R client signal must be transcoded into 1024B/1027B for rate reduction, the 64B/66B PCS receive process at the ingress interface further descrambles the bit-stream skipping sync headers and PCS lane alignment markers, and the 64B/66B PCS transmit process at the egress interface scrambles the bit-stream again skipping sync headers and PCS lane alignment markers, as shown in Figure E.1.

E.3.2 100GBASE-R client frame recovery

PCS lane alignment markers have the values shown in Table E.2 for 100GBASE-R signals which use PCS lane numbers 0-19.

The lane alignment markers transported over the OPU4 are distributed unchanged to the PCS lanes.

Table E.2 – PCS lane alignment marker format for 100GBASE-R

Lane Number	SH	Encoding $\{M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7\}$	Lane Number	SH	Encoding $\{M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7\}$
0	10	0xc1, 0x68, 0x21, BIP ₃ , 0x3e, 0x97, 0xde, BIP ₇	10	10	0xfd, 0x6c, 0x99, BIP ₃ , 0x02, 0x93, 0x66, BIP ₇
1	10	0x9d, 0x71, 0x8e, BIP ₃ , 0x62, 0x8e, 0x71, BIP ₇	11	10	0xb9, 0x91, 0x55, BIP ₃ , 0x46, 0x6e, 0xaa, BIP ₇
2	10	0x59, 0x4b, 0xe8, BIP ₃ , 0xa6, 0xb4, 0x17, BIP ₇	12	10	0x5c, 0xb9, 0xb2, BIP ₃ , 0xa3, 0x46, 0x4d, BIP ₇
3	10	0x4d, 0x95, 0x7b, BIP ₃ , 0xb2, 0x6a, 0x84, BIP ₇	13	10	0x1a, 0xf8, 0xbd, BIP ₃ , 0xe5, 0x07, 0x42, BIP ₇
4	10	0xf5, 0x07, 0x09, BIP ₃ , 0x0a, 0xf8, 0xf6, BIP ₇	14	10	0x83, 0xc7, 0xca, BIP ₃ , 0x7c, 0x38, 0x35, BIP ₇
5	10	0xdd, 0x14, 0xc2, BIP ₃ , 0x22, 0xeb, 0x3d, BIP ₇	15	10	0x35, 0x36, 0xcd, BIP ₃ , 0xca, 0xc9, 0x32, BIP ₇
6	10	0x9a, 0x4a, 0x26, BIP ₃ , 0x65, 0xb5, 0xd9, BIP ₇	16	10	0xc4, 0x31, 0x4c, BIP ₃ , 0x3b, 0xce, 0xb3, BIP ₇
7	10	0x7b, 0x45, 0x66, BIP ₃ , 0x84, 0xba, 0x99, BIP ₇	17	10	0xad, 0xd6, 0xb7, BIP ₃ , 0x52, 0x29, 0x48, BIP ₇
8	10	0xa0, 0x24, 0x76, BIP ₃ , 0x5f, 0xdb, 0x89, BIP ₇	18	10	0x5f, 0x66, 0x2a, BIP ₃ , 0xa0, 0x99, 0xd5, BIP ₇
9	10	0x68, 0xc9, 0xfb, BIP ₃ , 0x97, 0x36, 0x04, BIP ₇	19	10	0xc0, 0xf0, 0xe5, BIP ₃ , 0x3f, 0x0f, 0x1a, BIP ₇

E.4 Additions to Annex B transcoding for parallel 64B/66B clients

When OPUk is large enough for the serialized 66B block stream (e.g., for 100GBASE-R client signals into OPU4), the recovered client frames are adapted directly as per this annex.

When used in combination with the transcoding into 513B code blocks described in Annex B (e.g., for 40GBASE-R client signals into OPU3), this clause describes the additions to the Annex B transcoding process for transport of PCS lane alignment markers.

Ethernet path monitoring is the kind of behaviour that is desirable in the case where the Ethernet equipment and the OTN equipment are in different domains (e.g., customer and service provider) and from the standpoint of the Ethernet equipment. It is also the default behaviour which would result from the current mapping of 100GBASE-R where the 66B blocks would be mapped into the OPU4 container after management of skew. It may also be perceived as a transparency requirement that BIP-8 work end-to-end. Additional functionality as described below has to be built in to allow BIP-8 transparency for 40GBASE-R client signals.

PCS lane alignment markers are encoded together with 66B control blocks into the uppermost rows of the 513B code block shown in Figure B.3. The flag bit "F" of the 513B structure is 1 if the 513B structure contains at least one 66B control block or PCS lane alignment marker, and 0 if the 513B structure contains eight all-data 66B blocks.

The transcoding into 512B/513B must encode PCS lane alignment marker into a row of the structure shown in Figure B.3 as follows: The sync header of "10" is removed. The received M_0 , M_1 and M_2 bytes of the PCS alignment marker encodings as shown in Table E.1 are used to forward the lane number information. The first byte of the row will contain the structure shown in Figure B.4, with a CB-TYPE field of "0100". The POS field will indicate the position where the PCS lane alignment marker was received among the group of eight 66B codewords being encoded into this 513B block. The flag continuation bit "FC" will indicate whether any other 66B control blocks or PCS lane alignment markers are encoded into rows below this one in the 513B block. Beyond this first byte, the next four bytes of the row are populated with the received M_0 , M_1 , M_2 and ingress BIP₃ bytes of the PCS alignment marker encodings at the encoder. At the decoder, a PCS lane alignment marker will be generated in the position indicated by the POS field among any 66B all-data blocks contained in this 513B block, the sync header of "10" is generated followed by the received M_0 , M_1 and M_2 bytes, the egress BIP₃ byte, the bytes M_4 , M_5 and M_6 which are the bit-wise inverted M_0 , M_1 and M_2 bytes received at the decoder, and the egress BIP₇ byte which is the bit-wise inverted egress BIP₃ byte.

It will then be up to the Ethernet receiver to handle bit errors within the OTN section that might have altered the PCS alignment marker encodings (for details refer to clause 82.2.18.3 and Figure 82-11 in [IEEE 802.3]).

The egress BIP₃ and the egress BIP₇ bytes are calculated as described in clause E.4.1.

Figure E.2 below shows the transcoded lane marker format.

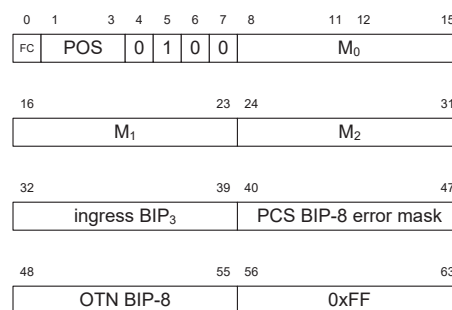


Figure E.2 – Transcoded lane marker format

E.4.1 BIP-8 transparency

The transcoding method used for 40GBASE-R is timing and PCS codeword transparent. In normal operation, the only aspects of the PCS encoded bitstream that are not preserved given the mapping described in annexes B, E and F are for one the scrambling, since the scrambler does not begin with a known state and multiple different encoded bitstreams can represent the same PCS encoded content, and secondly the BIP-8 value in the Ethernet path or more precisely, the bit errors that occur between the Ethernet transmitter and the ingress point of the OTN domain and within the OTN domain. The BIP-8 values can be preserved with the scheme described below. As the scrambling itself does not contain any information that has to be preserved, no effort has been made to synchronize the scrambler states between OTN ingress and OTN egress.

Unfortunately, since the BIP-8 is calculated on the scrambled bitstream, a simple transport of the BIP-8 across the OTN domain in the transcoded lane marker will not result in a BIP-8 value that is meaningful for detecting errors in the received, descrambled, transcoded, trans-decoded, and then rescrambled bit stream.

To preserve the bit errors between the Ethernet transmitter and the egress side of the OTN domain, the bit-error handling is divided into two processes, one that takes place at the OTN ingress side, or encoder, and one on the OTN egress side, or decoder.

At the OTN ingress an 8-bit error mask is calculated by generating the expected BIP-8 for each PCS lane and XORing this value with the received BIP-8. This error mask will have a "1" for each bit of the BIP-8 which is wrong, and a "0" for each bit which is correct. This value is shown as a PCS BIP-8 error mask in Figure E.2.

In the event no errors are introduced across the OTN (as an FEC protected network can be an essentially zero error environment), the PCS BIP-8 error mask can be used to adjust the newly calculated PCS BIP-8 at the egress providing a reliable indication of the number of errors that are introduced across the full Ethernet path. If errors are introduced across the OTN, this particular BIP-8 calculation algorithm will not see these errors.

To overcome this situation, a new BIP-8 per lane for the OTN section is introduced. In the following this new BIP-8 will be identified as OTN BIP-8 in order to distinguish it from the PCS BIP-8.

It should be noted that the term OTN BIP-8 does not refer to and should not be confused with the BIP-8 defined in the OTUk overhead (byte SM[2]).

The OTN BIP-8 is calculated similarly to the PCS BIP-8 as described in clause 82.2.8 of [IEEE 802.3] with the exception that the calculation will be done over unscrambled PCS lane data, the original received lane alignment marker, after error control block insertion and before transcoding. Figure E.2 shows the byte location of the OTN BIP-8 in the transcoded lane marker.

The transcoded lane marker is transmitted together with the transcoded data blocks over the OTN section as defined in Annex B. At the OTN egress after trans-decoding and before scrambling, the ingress alignment marker is recreated using M_0 , M_1 , M_2 and ingress BIP₃ of the transcoded alignment marker followed by the bit-wise inversion of these bytes. This recreated alignment marker together with the trans-decoded and unscrambled data blocks is used to calculate the expected OTN BIP-8 for each PCS lane (refer to clause 82.2.8 of [IEEE 802.3]). The expected value will be XORed with the received OTN BIP-8. This error mask will have a "1" for each bit of the OTN BIP-8 which is wrong, and a "0" for each bit which is correct.

The egress BIP₃ for each PCS lane is calculated over the trans-decoded and scrambled data blocks including the trans-decoded alignment marker (refer to clause E.4) following the process depicted in clause 82.2.8 of [IEEE 802.3].

The egress BIP₃ is then adjusted for the errors that occurred up to the OTN egress by first XORing with the PCS BIP-8 error mask and then XORing with the OTN BIP-8 error mask. This combined error mask will be used to compute the number of BIP errors when used for non-intrusive monitoring.

The BIP₇ is created by bit-wise inversion of the adjusted BIP₃.

E.4.2 Errors detected by mapper

Errors encountered before the mapper, such as loss of client signal on any physical lane of the interface, will result in the insertion of an Ethernet LF sequence ordered set prior to this process. The same action should be taken as a result of failure to achieve 66B block lock on any PCS lane, failure to achieve lane alignment marker framing on each PCS lane, or failure to de-skew because the skew exceeds the buffer available for de-skew.

An invalid 66B block will be converted to an error control block before transcoding. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10", is not a valid PCS lane alignment marker and has a control block type field which does not appear in Figure B.2 or has one of the values 0x2d, 0x33, 0x66, or 0x55 which are not used for 40GBASE-R or 100GBASE-R. An error control block has sync bits of "10", a block type code of 0x1e, and 8 seven-bit/E/error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

E.4.3 Errors detected by de-mapper

Several mechanisms will be employed to reduce the probability that the de-mapper constructs erroneous parallel 64B/66B encoded data at the egress if bit errors have corrupted. Since detectable corruption normally means that the proper order of 66B blocks to construct at the de-mapper cannot be reliably determined, if any of these checks fail, the de-mapper will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Mechanisms for improving the robustness and for 513B block lock including PCS lane alignment markers are discussed in Annex F.

Annex F

Improved robustness for mapping of 40GBASE-R into OPU3 using 1027B code blocks

(This annex forms an integral part of this Recommendation.)

F.1 Introduction

When a parallel 40GBASE-R signal is transcoded as per Annexes B and E and directly mapped into OPU3 without GFP framing, another method is needed to locate the start of 513B blocks and to provide protection to prevent bit errors creating an unacceptable increase in mean time to false packet acceptance (MTTFPA).

F.2 513B code block framing and flag bit protection

The mapping of 513B code blocks into OPU3 requires a mechanism for locating the start of the code blocks. A mechanism is also needed to protect the flag bit, whose corruption could cause data to be erroneously interpreted as control and vice versa.

Both of these requirements can be addressed by providing parity across the flag bits of two 513B blocks produced from the transcoding of Annex B.

Figure F.1 illustrates the flag parity bit across two 513B blocks. This creates a minimum two-bit Hamming distance between valid combinations of flag bits.

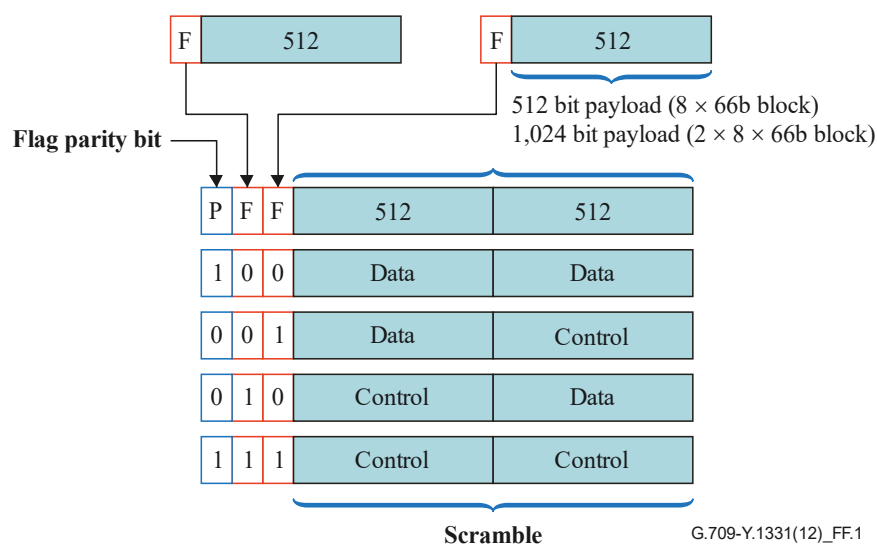


Figure F.1 – Flag parity bit on two 513B blocks (1027B code)

The flag parity bit creates a sequence that can be used for framing to locate the 513B blocks in a stream of bits. The state diagram of Figure 49-12 of [IEEE 802.3] is applied to locate a 3-bit pattern appearing once per 1027 bits (rather than a 2-bit pattern appearing once per 66 bits) where four out of eight 3-bit sequences (rather than two out of four two-bit values as used in IEEE 802.3) match the pattern. The additional step required is to scramble the non-flag bits so that the legal sequences of these bits are not systematically mimicked in the data itself. The scrambler to be used for this purpose is the Ethernet self-synchronous scrambler using the polynomial $G(x) = 1 + x^{39} + x^{58}$.

At the de-mapper, invalid flag bit parity will cause both of the 513B blocks across which the flag bit parity applies to be decoded as 8×2 66B error control blocks ("10" sync header, control block type 0x1e, followed by eight 7-bit/E/control characters).

F.3 66B block sequence check

Bit error corruption of the position or flag continuation bits could cause 66B blocks to be de-mapped from 513B code blocks in the incorrect order. Additional checks are performed to prevent that this results in incorrect packet delineation. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit/E/control characters).

Other checks are performed to reduce the probability that invalid data is delivered at the egress in the event that bit errors have corrupted any of the POS fields or flag continuation bits "FC".

If flag bit "F" is 1 (i.e., the 513B block includes at least one 64B/66B control block), for the rows of the table up until the first one with a flag continuation bit of zero (the last one in the block), it is verified that no two 66B control blocks or lane alignment markers within that 513B block have the same value in the POS field, and further, that the POS field values for multiple control or lane alignment rows are in ascending order, which will always be the case for a properly constructed 513B block. If this check fails, the 513B block is decoded into eight 66B error control blocks.

The next check is to ensure that the block sequence corresponds to well-formed packets, which can be done according to the state diagram in Figures F.2 and F.3. This check will determine if 66B blocks are in an order that does not correspond to well-formed packets, e.g., if during an IPG an all-data 66B block is detected without first seeing a control block representing packet start, or if during a packet a control/idle block is detected without first seeing a control block representing packet termination, control blocks have likely been misordered by corruption of either the POS bits or a flag continuation bit. Failure of this check will cause the 513B block to be decoded as eight 66B error control blocks. Note that PCS lane alignment markers are accepted in either state and do not change state as shown in Figure F.3.

The sequence of PCS lane alignment markers is also checked at the decoder. For an interface with p PCS lanes, the PCS lane alignment markers for lanes 0 to $p-1$ will appear in a sequence, followed by $16383 \times p$ non-lane-marker 66B blocks, followed by another group of PCS lane alignment markers. A counter is maintained at the decoder to keep track of when the next group of lane alignment markers is expected. If, in the process of decoding lane alignment markers from a 513B block, a lane alignment marker is found in a position where it is not expected, or a lane alignment marker is missing in a position where it would have been expected, the entire 513B block is decoded as eight 66B error control blocks as shown in Figures F.2, F.3, and F.4.

F.3.1 State diagram conventions

The body of this clause is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

The notation used in the state diagrams follows the conventions of clause 21.5 of [IEEE 802.3]. State diagram timers follow the conventions of clause 14.2.3.2 of [IEEE 802.3]. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

F.3.2 State variables

F.3.2.1 Constants

EBLOCK_T<65:0>

66-bit vector to be sent to the PCS containing /E/ in all the eight character locations.

Mi<65:0>

66-bit vector containing the trans-decoded alignment marker of i-th PCS lane ($0 < i \leq p$). ($p=4$ for 40GBASE-R, and $p=20$ for 100GBASE-R).

F.3.2.2 Variables

1027B_block_lock

Indicates the state of the block_lock variable when the state diagram of Figure 49-12 of [IEEE 802.3] is applied to locate a 3-bit pattern appearing once per 1027 bits (rather than a 2-bit pattern appearing once per 66 bits) as described in clause F.2. Set true when sixty-four contiguous 1027-bit blocks are received with valid 3-bit patterns, set false when sixteen 1027-bit blocks with invalid 3-bit patterns are received before sixty-four valid blocks.

1027B_high_ber

Indicates a Boolean variable when the state diagram of Figure 49-13 of [IEEE 802.3] is applied to count invalid 3-bit sync headers of 1027-bit blocks (rather than 2-bit sync headers of 66-bit blocks) within the current 250 μ s (rather than 125 μ s). Set true when the ber_cnt exceeds 8 (rather than 16) indicating a bit error ratio $>10^{-4}$.

Mseq_violation

Indicates a Boolean variable that is set and latched in each rx513_raw<527:0> PCS lane alignment marker cycle based on the PCS lane marker position and order. It is true if the unexpected marker sequence is detected and false if not.

POS_violation

A Boolean variable that is set in each rx513_raw<527:0> based on the POS field values for rx_tcd<65:0>. It is true if the two or more have the same POS values or if they are not in ascending order, and false if their POS values are in ascending order.

reset

A Boolean variable that controls the resetting of the PCS. It is true whenever a reset is necessary including when reset is initiated from the MDIO, during power on, and when the MDIO has put the PCS into low-power mode.

Rx513_coded<512:0>

A vector containing the input to the 512B/513B decoder.

rx513_raw<527:0>

A vector containing eight successive 66-bit vectors (tx_coded).

rx_tcd<65:0>

A 66-bit vector transcode-decoded from a 513-bit block following the rules shown in Figure B.5.

seq_violation

A Boolean variable that is set in each rx513_raw<527:0> based on the sequence check on an rx_tcd<65:0> stream. It is true if the unexpected sequence is detected and false if not.

F.3.2.3 Functions

DECODE(rx513_coded<512:0>)

Decodes the 513-bit vector returning rx513_raw<527:0> which is sent to the client interface. The DECODE function shall decode the block as specified in Figure F.2.

$R_BLOCK_TYPE = \{C, S, T, D, E, M\}$

This function classifies each 66-bit rx_tcd vector as belonging to one of the six types depending on its contents.

Values: C, S, T, and D are defined in clause 49.2.13.2.3 of [IEEE 802.3].

M: the vector contains a sync header of 10 and is recognized as a valid PCS lane alignment marker by using the state machine shown in Figure F.3.

E: the vector does not meet the criteria for any other value.

$R_TYPE(rx_tcd\langle 65:0 \rangle)$

Returns the R_BLOCK_TYPE of the $rx_tcd\langle 65:0 \rangle$ bit vector.

R_TYPE_NEXT

Prescient end of packet check function. It returns the R_BLOCK_TYPE of the rx_tcd vector immediately following the current rx_tcd vector.

F.3.2.4 Counters

cnt

Count up to a maximum of p of the number of PCS lanes.

F.3.3 State diagrams

The receive state machine for a series of 513-bit blocks shown in Figure F.2 determines whether the 513-bit block contains valid eight 66-bit blocks or not.

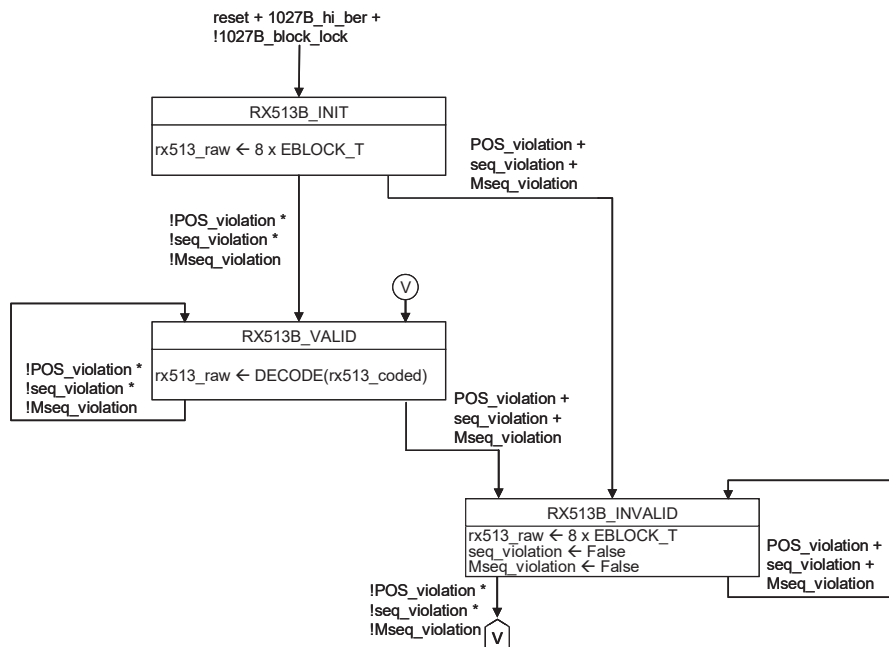


Figure F.2 – Receive state machine for the 512B/513B code blocks including lane alignment markers

The trans-decode state machine for a series of 66-bit blocks shown in Figure F.3 checks the block type sequence of recovered 66-bit blocks.

The PCS lane alignment marker state machine for a series of 66-bit blocks shown in Figure F.4 detects the alignment markers every $p \times 16384$ blocks and checks whether the marker is in ascendant order or not.

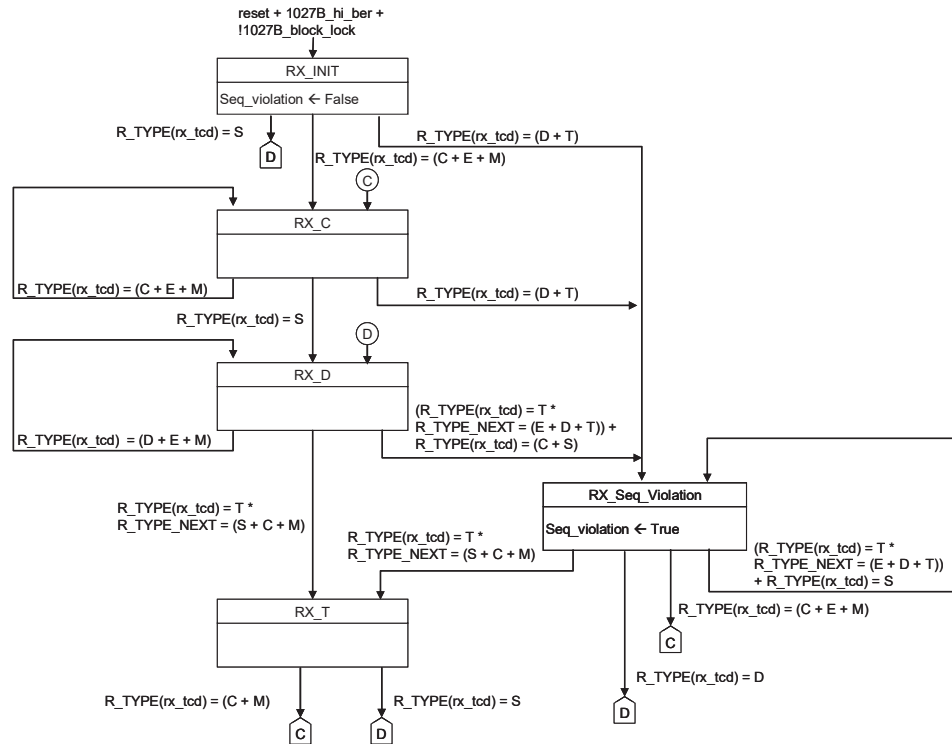


Figure F.3 – Trans-decode state machine for the 64B/66B code blocks including the lane alignment markers

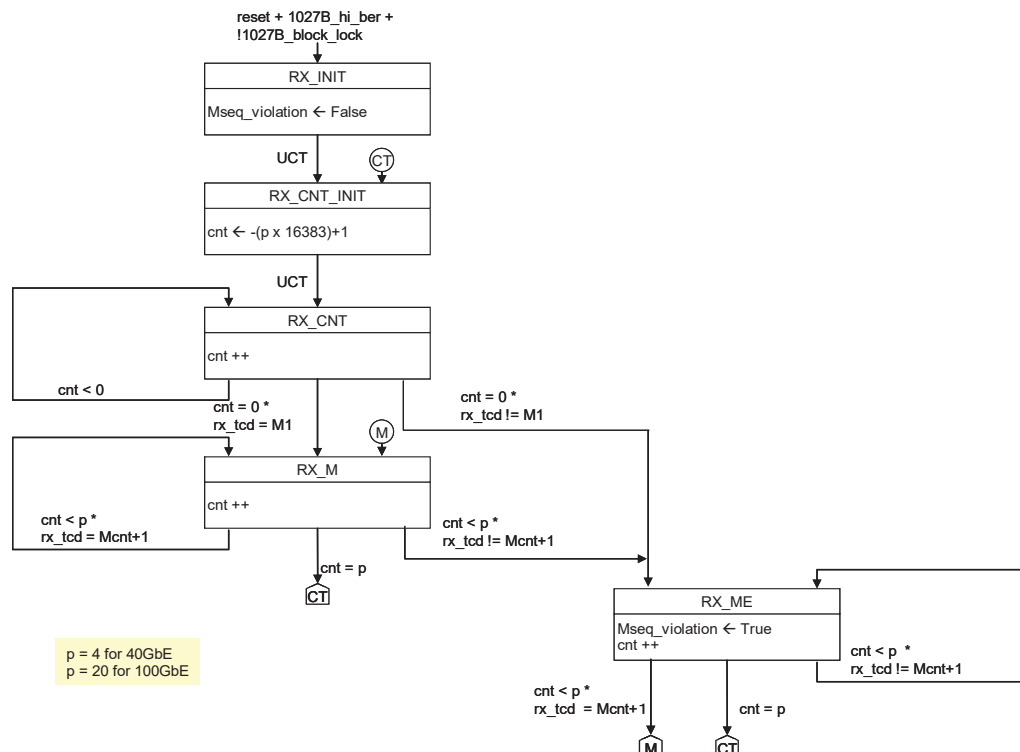


Figure F.4 – Receive state machine for the lane alignment markers

Annex G

Mapping ODU0 into a low latency OTU0 (OTU0LL)

(This annex forms an integral part of this Recommendation.)

G.1 Introduction

Within the optical transport network client signals with a bit rate up to 1.25 Gbit/s are transported within an ODU0 and the ODU0 is transported within a server ODU_k and OTU_k ($k = 1,2,3,4$). This annex specifies a low latency 1.25G OTU0 (OTU0LL) frame format in which one ODU0 is transported which carries a client (e.g., 1G Ethernet) signal using an [ITU-T G.698.3] application code. This OTU0LL may be used at the edge of the optical transport network.

G.2 Optical transport unit 0 low latency (OTU0LL)

The OTU0LL conditions the ODU0 for transport over a multi-vendor optical network interface at the edge of the optical transport network. The OTU0LL frame structure, including the OTU0LL FEC is completely standardized. The optical aspects of the multi-vendor optical network interface at the edge of the optical transport network are outside the scope of this Recommendation.

NOTE 1 – Transport of the OTU0LL over the SOTU, MOTU, SOTU_m and MOTU_m interfaces specified in this Recommendation is not supported.

NOTE 2 – An ODU0 which is transported within an OTU0LL may be passed through the OTN and terminated at the far end edge of the OTN.

G.2.1 OTU_k frame structure

The OTU0LL frame structure is based on the ODU0 frame structure and extends it with a distributed forward error correction (FEC) as shown in Figure G.1. Sixteen times 16 columns are added to the ODU0 frame for the FEC and the reserved overhead bytes in row 1, columns 8 to 14 of the ODU0 overhead are used for an OTU0LL specific overhead, resulting in an octet-based block frame structure with four rows and 4080 columns. The MSB in each octet is bit 1, the LSB is bit 8.

The OTU0LL overhead is the same as the OTU_k overhead.

The bit rate of the OTU0LL signal is $255/239 \times 1\,244\,160$ kbit/s ($\sim 1\,327\,451.046$ kbit/s). The frame period of the OTU0LL signal is approximately 98.354 μ s.

The sixty-four 16-byte RS(255,239) FEC fields in the OTU0LL frame contain the Reed-Solomon RS(255,239) FEC code. Each RS(255,239) FEC is computed over the previous 239 OTU0LL bytes. Transmission of the OTU0LL FEC is mandatory.

NOTE – The distribution of the RS(255,239) FEC fields over the OTU0LL frame minimizes the transfer delay introduced by the processing of this FEC and the number of codecs to compute this FEC.

The RS(255,239) FEC code shall be computed as specified in Annex A with the notion that each FEC is computed over the previous 239 OTU0LL bytes instead of over a sub-row as described in this annex for the case of an OTU_k ($k=1,2,3,4$).

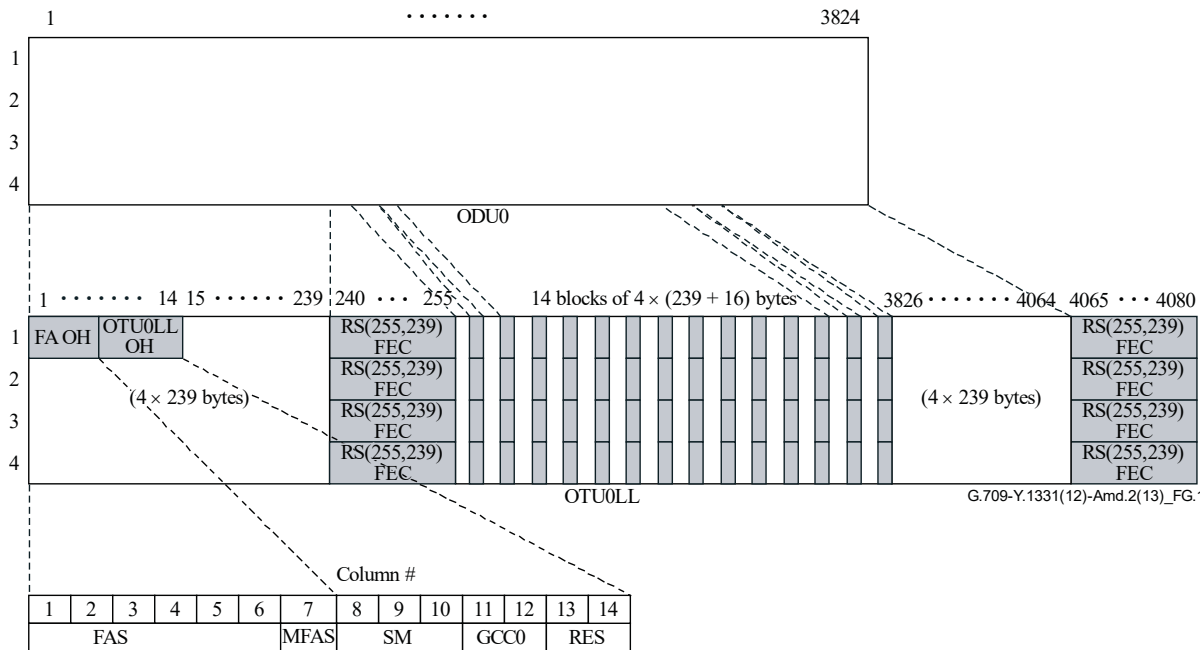


Figure G.1 – OTU0LL frame structure, overhead and ODU0 mapping

The transmission order of the bits in the OTU0LL frame is left to right, top to bottom, and MSB to LSB (see Figure G.2).

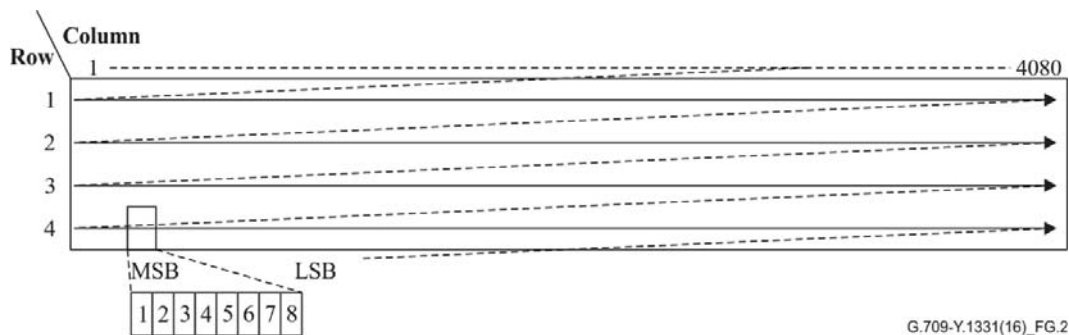


Figure G.2 – Transmission order of the OTU0LL frame bits

G.2.2 Scrambling

The OTU0LL signal must have sufficient bit timing content at the NNI. A suitable bit pattern, which prevents a long sequence of "1"s or "0"s, is provided by using a scrambler.

The operation of the scrambler shall be functionally identical to that of a frame synchronous scrambler of sequence length 65535 operating at the OTU0LL rate.

The generating polynomial shall be $1 + x + x^3 + x^{12} + x^{16}$. Figure G.3 shows a functional diagram of the frame synchronous scrambler.

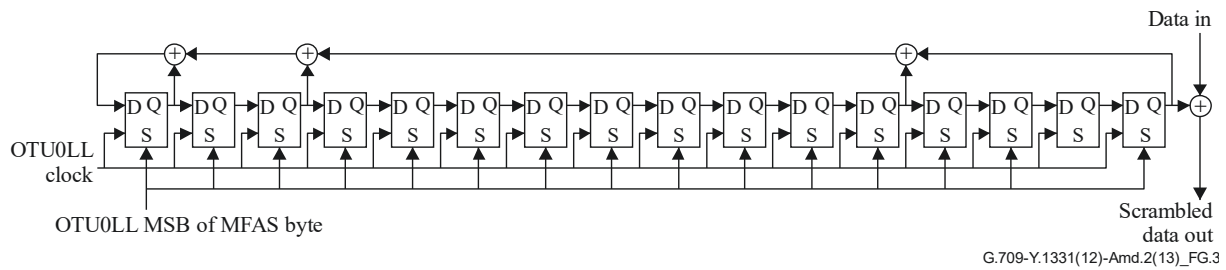


Figure G.3 – Frame synchronous scrambler

The scrambler shall be reset to "FFFF" (HEX) on the most significant bit of the byte following the last framing byte in the OTU0LL frame, i.e., the MSB of the MFAS byte. This bit, and all subsequent bits to be scrambled shall be added modulo 2 to the output from the x^{16} position of the scrambler. The scrambler shall run continuously throughout the complete OTU0LL frame. The framing bytes (FAS) of the OTU0LL overhead shall not be scrambled.

Scrambling is performed after FEC computation and insertion into the OTU0LL signal.

Annex H

OTUCn sub rates (OTUCn-M)

(This annex forms an integral part of this Recommendation.)

H.1 Introduction

This annex defines a subrate OTUCn, note that subrates are not defined for the OPUCn and ODUcN. Additional descriptive text, examples and applications are provided in Appendix XIII.

The OPUCn, ODUcN and OTUCn are defined in terms of $n \times 100$ Gbit/s signals. For the MOTUm interface it may be advantageous to carry an OTUCn with a bit rate that is not constrained to be an integer multiple of 100 Gbit/s. This OTU is described as an OTUCn-M.

H.2 OTUCn-M frame format

The OTUCn-M frame is a type of OTUCn frame which contains n instances of OTUC, ODUc and OPUC overhead and $M \times 5$ Gbit/s OPUCn tributary slots. If a particular value of M is not indicated, the frame contains $20 \times n$ tributary slots. See Appendix XIII for examples.

An OTUCn-M framed signal can be carried over MOTUm interfaces. Values of M supported and the subset of the $20n \times 5$ Gbit/s OPUCn tributary slots that are included in the OTUCn-M on such a MOTUm interface are vendor specific and outside the scope of this recommendation.

The OPU tributary slots that are included in the OTUCn-M structure will have their availability bit in the OPUCn MSI overhead set to "available" (see 20.4.1.1). The OPU tributary slots which might not be carried transparently by the OTUCn-M structure will have their availability bit in the OPUCn MSI overhead set to "unavailable" (see 20.4.1.1). The data in tributary slots marked as unavailable must be set to "all 0".

Appendix I

Range of stuff ratios for asynchronous mappings of CBR2G5, CBR10G, and CBR40G clients with ± 20 ppm bit-rate tolerance into OPuK, and for asynchronous multiplexing of ODUj into ODUk ($k > j$)

(This appendix does not form an integral part of this Recommendation.)

Clause 17.2 describes asynchronous and bit-synchronous mappings of CBR2G5, CBR10G, and CBR40G clients with ± 20 ppm bit-rate tolerance into ODU1, 2, and 3, respectively. Clause 19 describes asynchronous mapping (multiplexing) of ODUj into ODUk ($k > j$). For asynchronous CBR client mappings, any frequency difference between the client and local OPuK server clocks is accommodated by the $+1/0/-1$ justification scheme. For asynchronous multiplexing of ODUj into ODUk ($k > j$), any frequency difference between the client ODUj and local OPuK server clocks is accommodated by the $+2/+1/0/-1$ justification scheme. The OPuK payload, ODUk, and OTUk bit rates and tolerances are given in clause 7.3. The ODU1, ODU2, and ODU3 rates are 239/238, 239/237, and 239/236 times 2 488 320 kbit/s, 9 953 280 kbit/s, and 39 813 120 kbit/s, respectively. The ODUk bit-rate tolerances are ± 20 ppm. This appendix shows that each justification scheme can accommodate these bit rates and tolerances for the respective mappings, and also derives the range of justification (stuff) ratio for each mapping.

The $+1/0/-1$ mapping in clause 17.2 provides for one positive justification opportunity (PJO) and one negative justification opportunity (NJO) in each ODUk frame. The $+2/+1/0/-1$ mapping in clause 19 provides for 2 PJOs and one NJO in each ODUk frame. For the case of ODU multiplexing (i.e., the latter case), the ODUj being mapped will get only a fraction of the full payload capacity of the ODUk. There can be, in general, a number of fixed stuff bytes per ODUj or CBR client. Note that in both mapping cases, there is one stuff opportunity in every ODUk frame. For mapping of a CBR client into ODUk, the CBR client is allowed to use all the stuff opportunities (because only one CBR client signal is mapped into an ODUk). However, for mapping ODUj into ODUk ($k > j$), the ODUj can only use $1/2$ (ODU0 into ODU1), $1/4$ (ODU1 into ODU2 or ODU2 into ODU3) or $1/16$ (ODU1 into ODU3) of the stuff opportunities. The other stuff opportunities are needed for the other clients being multiplexed into the ODUk.

Traditionally, the justification ratio (stuff ratio) for purely positive justification schemes is defined as the long-run average fraction of justification opportunities for which a justification is done (i.e., for a very large number of frames, the ratio of the number of justifications to the total number of justification opportunities). In the $+1/0/-1$ scheme, positive and negative justifications must be distinguished. This is done by using different algebraic signs for positive and negative justifications. With this convention, the justification ratio can vary at most (for sufficiently large frequency offsets) from -1 to $+1$ (in contrast to a purely positive justification scheme, where the justification ratio can vary at most from 0 to 1). In the case of ODUk multiplexing, the justification ratio is defined relative to the stuff opportunities available for the client in question. Then, the justification ratio can vary (for sufficiently large frequency offsets) from -1 to $+2$. (If the justification ratio were defined relative to all the stuff opportunities for all the clients, the range would be $-1/2$ to $+1$ for multiplexing ODU0 into ODU1, $-1/4$ to $+1/2$ for multiplexing ODU1 into ODU2 and ODU2 into ODU3, and $-1/16$ to $+1/8$ for multiplexing ODU1 into ODU3.)

Let α represent the justification ratio ($-1 \leq \alpha \leq 1$ for CBR client into ODUk mapping; $-2 \leq \alpha \leq 1$ for ODUj into ODUk mapping ($k > j$)), and use the further convention that positive α will correspond to negative justification and negative α to positive justification (the reason for this convention is explained below).

Define the following notation (the index j refers to the possible ODU $_j$ client being mapped, and the index k refers to the ODU $_k$ server layer into which the ODU $_j$ or CBR client is mapped):

- N = number of fixed stuff bytes in the OPU $_k$ payload area associated with the client in question (note that this is not the total number of fixed stuff bytes if multiple clients are being multiplexed)
- S = nominal STM-N or ODU $_j$ client rate (bytes/s)
- T = nominal ODU $_k$ frame period(s)
- y_c = client frequency offset (fraction)
- y_s = server frequency offset (fraction)
- p = fraction of OPU $_k$ payload area available to this client
- N_f = average number of client bytes mapped into an ODU $_k$ frame, for the particular frequency offsets (averaged over a large number of frames)

Then N_f is given by:

$$N_f = ST \frac{1 + y_c}{1 + y_s} \quad (\text{I-1})$$

For frequency offsets small compared to 1, this may be approximated:

$$N_f = ST(1 + y_c - y_s) \equiv ST\beta \quad (\text{I-2})$$

The quantity $\beta - 1$ is the net frequency offset due to client and server frequency offset.

Now, the average number of client bytes mapped into an ODU $_k$ frame is also equal to the total number of bytes in the payload area available to this client (which is $(4)(3808)p = 15232p$), minus the number of fixed stuff bytes for this client (N), plus the average number of bytes stuffed for this client over a very large number of frames. The latter is equal to the justification ratio α multiplied by the fraction of frames p corresponding to justification opportunities for this client. Combining this with equation I-1 produces:

$$ST\beta + \alpha p + 15232p - N \quad (\text{I-3})$$

In equation I-3, a positive α corresponds to more client bytes mapped into the ODU $_k$, on average. As indicated above, this corresponds to negative justification. This sign convention is used so that α enters in equation I-3 with a positive sign (for convenience).

Equation I-3 is the main result. For mapping STM-N (CBR clients) into ODU $_k$, the quantity p is 1.

The range of stuff ratio may now be determined for mapping STM-N or ODU $_j$ clients into ODU $_k$, using equation I-3. In what follows, let R_{16} be the STM-16 rate, i.e., $2.48832 \text{ Gbit/s} = 3.1104 \times 10^8 \text{ bytes/s}$.

Asynchronous mapping of CBR2G5 (2 488 320 kbit/s) signal into ODU1

The nominal client rate is $S = R_{16}$. The nominal ODU1 rate is $(239/238)S$ (see clause 7.3). But the nominal ODU1 rate is also equal to $(4)(3824)/T$. Then:

$$ST = (4)(3824) \frac{238}{239} = 15232 \quad (\text{I-4})$$

Inserting this into equation I-3, and using the fact that $N = 0$ (no fixed stuff bytes) for this mapping produces:

$$\alpha = 15232(\beta - 1) \quad (\text{I-5})$$

Since the ODUk and client frequency tolerances are each ± 20 ppm, β ranges from 0.99996 to 1.00004. Using this in equation I-5 gives as the range of α :

$$-0.60928 \leq \alpha \leq +0.60928 \quad (\text{I-6})$$

Asynchronous mapping of CBR10G (9 953 280 kbit/s) signal into ODU2

The nominal client rate is $S = 4R_{16}$. The nominal ODU2 rate is $(239/237)S$ (see clause 7.3). But the nominal ODU2 rate is also equal to $(4)(3824)/T$. Then:

$$ST = (4)(3824) \frac{237}{239} = 15168 \quad (\text{I-7})$$

Inserting this into equation I-3, and using the fact that $N = 64$ (number of fixed stuff bytes) for this mapping produces:

$$\alpha = 15168\beta + 64 - 15232 = 15168(\beta - 1) \quad (\text{I-8})$$

As before, the ODUk and client frequency tolerances are ± 20 ppm, and β ranges from 0.99996 to 1.00004. Using this in equation I-8 gives as the range of α :

$$-0.60672 \leq \alpha \leq +0.60672 \quad (\text{I-9})$$

Asynchronous mapping of CBR40G (39 813 120 kbit/s) signal into ODU3

The nominal client rate is $S = 16R_{16}$. The nominal ODU3 rate is $(239/236)S$ (see clause 7.3). But the nominal ODU3 rate is also equal to $(4)(3824)/T$. Then:

$$ST = (4)(3824) \frac{236}{239} = 15104 \quad (\text{I-10})$$

Inserting this into equation I-3, and using the fact that $N = 128$ (number of fixed stuff bytes) for this mapping produces:

$$\alpha = 15104\beta + 128 - 15232 = 15104(\beta - 1) \quad (\text{I-11})$$

As before, the ODUk and client frequency tolerances are ± 20 ppm, and β ranges from 0.99996 to 1.00004. Using this in equation I-11 gives as the range of α :

$$-0.60416 \leq \alpha \leq +0.60416 \quad (\text{I-12})$$

ODU1 into ODU2 multiplexing

The ODU1 nominal client rate is (see clause 7.3):

$$S = \frac{239}{238} R_{16} \quad (\text{I-13})$$

The ODU2 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{237}(4R_{16})} \quad (\text{I-14})$$

The fraction p is 0.25. Inserting into equation I-3 produces:

$$\frac{239}{238} R_{16} \frac{(3824)(4)}{\frac{239}{237}(4R_{16})} \beta = \frac{\alpha}{4} + 3808 - N \quad (\text{I-15})$$

Simplifying and solving for α produces:

$$\alpha = \frac{237}{238}(15296)\beta + 4N - 15232 \quad (\text{I-16})$$

Now let $\beta = 1 + y$, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned} \alpha &= \frac{237}{238}(15296) - 15232 + 4N + \frac{237}{238}(15296)y \\ &= 4N - 0.2689076 + 15231.731092y \end{aligned} \quad (\text{I-17})$$

The number of fixed stuff bytes N is zero, as given in clause 19.5.1. The client and mapper frequency offsets are in the range ± 20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ± 40 ppm. Inserting these values into equation I-17 gives for the range for α :

$$\begin{aligned} \alpha &= 0.340362 & \text{for } y = +40 \text{ ppm} \\ \alpha &= -0.268908 & \text{for } y = 0 \text{ ppm} \\ \alpha &= -0.878177 & \text{for } y = -40 \text{ ppm} \end{aligned} \quad (\text{I-18})$$

In addition, stuff ratios of -2 and $+1$ are obtained for frequency offsets of -113.65 ppm and 83.30 ppm, respectively. The range of frequency offset that can be accommodated is approximately 197 ppm. This is 50% larger than the range that can be accommodated by a $+1/0/-1$ justification scheme (see above), and is due to the additional positive stuff byte.

ODU2 into ODU3 multiplexing

The ODU2 nominal client rate is (see clause 7.3):

$$S = \frac{239}{237}(4R_{16}) \quad (\text{I-19})$$

The ODU3 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{236}(16R_{16})} \quad (\text{I-20})$$

The fraction p is 0.25 . Inserting into equation I-3 produces:

$$\frac{239}{237}4R_{16} \frac{(3824)(4)}{\frac{239}{236}(16R_{16})} \beta = \frac{\alpha}{4} + 3808 - N \quad (\text{I-21})$$

Simplifying and solving for α produces:

$$\alpha = \frac{236}{237}(15296)\beta + 4N - 15232 \quad (\text{I-22})$$

As before, let $\beta = 1 + y$, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned} \alpha &= \frac{236}{237}(15296) - 15232 + 4N + \frac{236}{237}(15296)y \\ &= 4N - 0.5400844 + 15231.459916y \end{aligned} \quad (\text{I-23})$$

The number of fixed stuff bytes N is zero, as given in clause 19.5.3. The client and mapper frequency offsets are in the range ± 20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ± 40 ppm. Inserting these values into equation I-23 gives for the range for α :

$$\begin{aligned}\alpha &= 0.0691740 & \text{for } y = +40 \text{ ppm} \\ \alpha &= -0.5400844 & \text{for } y = 0 \text{ ppm} \\ \alpha &= -1.149343 & \text{for } y = -40 \text{ ppm}\end{aligned}\tag{I-24}$$

In addition, stuff ratios of -2 and $+1$ are obtained for frequency offsets of -95.85 ppm and 101.11 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 197 ppm, which is 50% larger than the range that can be accommodated by a $+1/0/-1$ justification scheme (see above) due to the additional positive stuff byte.

ODU1 into ODU3 multiplexing

The ODU1 nominal client rate is (see clause 7.3):

$$S = \frac{239}{238}(R_{16})\tag{I-25}$$

The ODU3 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{236}(16R_{16})}\tag{I-26}$$

The fraction p is 0.0625. Inserting into equation I-3 produces:

$$\frac{239}{238}R_{16} \frac{(3824)(4)}{\frac{239}{236}(16R_{16})} \beta = \frac{\alpha}{16} + 952 - N\tag{I-27}$$

Simplifying and solving for α produces:

$$\alpha = \frac{236}{238}(15296)\beta + 16N - 15232\tag{I-28}$$

As before, let $\beta = 1 + y$, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned}\alpha &= \frac{236}{238}(15296) - 15232 + 16N + \frac{236}{238}(15296)y \\ &= 16N - 64.5378151 + 15167.462185y\end{aligned}\tag{I-29}$$

The total number of fixed stuff bytes in the ODU3 payload is 64, as given in clause 19.5.2; the number for one ODU1 client, N , is therefore 4. The client and mapper frequency offsets are in the range ± 20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ± 40 ppm. Inserting these values into equation I-29 gives for the range for α :

$$\begin{aligned}\alpha &= 0.0688834 & \text{for } y = +40 \text{ ppm} \\ \alpha &= -0.5378151 & \text{for } y = 0 \text{ ppm} \\ \alpha &= -1.144514 & \text{for } y = -40 \text{ ppm}\end{aligned}\tag{I-30}$$

In addition, stuff ratios of -2 and $+1$ are obtained for frequency offsets of -96.40 ppm and 101.39 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 197 ppm, which is 50% larger than the range that can be accommodated by a $+1/0/-1$ justification scheme (see above) due to the additional positive stuff byte.

ODU0 into ODU1 multiplexing

The ODU0 nominal client rate is (see clause 7.3):

$$S = \frac{1}{2}(R_{16}) \quad (\text{I-31})$$

The ODU1 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{238}(R_{16})} \quad (\text{I-32})$$

The fraction p is 0.5 . Inserting into equation I-3 produces:

$$\frac{1}{2}R_{16} \frac{(3824)(4)}{\frac{239}{238}(R_{16})} \beta = \frac{\alpha}{2} + 7616 - N \quad (\text{I-33})$$

Simplifying and solving for α produces:

$$\alpha = \frac{238}{239}(15296)\beta + 2N - 15232 \quad (\text{I-34})$$

As before, let $\beta = 1 + y$, where y is the net frequency offset (and is very nearly equal to $y_c - y_s$ for client and server frequency offset small compared to 1). Then:

$$\begin{aligned} \alpha &= \frac{238}{239}(15296) - 15232 + 2N + \frac{238}{239}(15296)y \\ &= 2N + 15232y \end{aligned} \quad (\text{I-35})$$

The total number of fixed stuff bytes N is zero, as given in clause 19.5.4. The client and mapper frequency offsets are in the range ± 20 ppm, as given in clause 7.3. Then, the net frequency offset y is in the range ± 40 ppm. Inserting these values into equation I-35 gives for the range for α :

$$\begin{aligned} \alpha &= 0.6092800 & \text{for } y = +40 \text{ ppm} \\ \alpha &= 0.0000000 & \text{for } y = 0 \text{ ppm} \\ \alpha &= -0.6092800 & \text{for } y = -40 \text{ ppm} \end{aligned} \quad (\text{I-36})$$

In addition, stuff ratios of -2 and $+1$ are obtained for frequency offsets of -130 ppm and 65 ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately 195 ppm.

Appendix II

Examples of functionally standardized OTU frame structures

(This appendix does not form an integral part of this Recommendation.)

This appendix provides examples of functionally standardized OTU frame structures. These examples are for illustrative purposes and by no means imply a definition of such structures. The completely standardized OTUk frame structure as defined in this Recommendation is shown in Figure II.1. Functionally standardized OTUkV frame structures will be needed to support, e.g., alternative FEC. Examples of OTUkV frame structures are:

- OTUkV with the same overhead byte allocation as the OTUk, but use of an alternative FEC as shown in Figure II.2;
- OTUkV with the same overhead byte allocation as the OTUk, but use of a smaller, alternative FEC code and the remainder of the OTUkV FEC overhead area filled with fixed stuff as shown in Figure II.3;
- OTUkV with a larger FEC overhead byte allocation as the OTUk, and use of an alternative FEC as shown in Figure II.4;
- OTUkV with no overhead byte allocation for FEC as shown in Figure II.5;
- OTUkV with a different frame structure than the OTUk frame structure, supporting a different OTU overhead (OTUkV overhead and OTUkV FEC) as shown in Figure II.6;
- OTUkV with a different frame structure than the OTUk frame structure, supporting a different OTU overhead (OTUkV overhead) and with no overhead byte allocation for FEC as shown in Figure II.7.

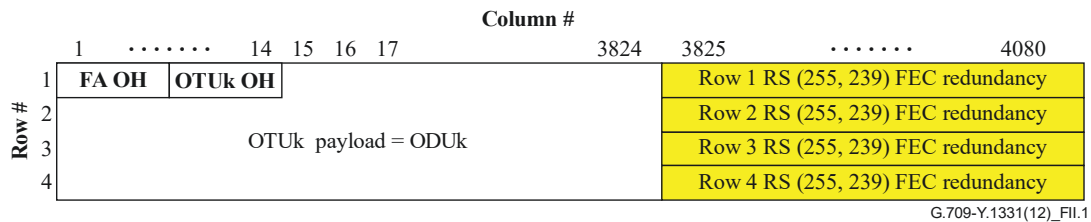


Figure II.1 – OTUk (with RS(255,239) FEC)

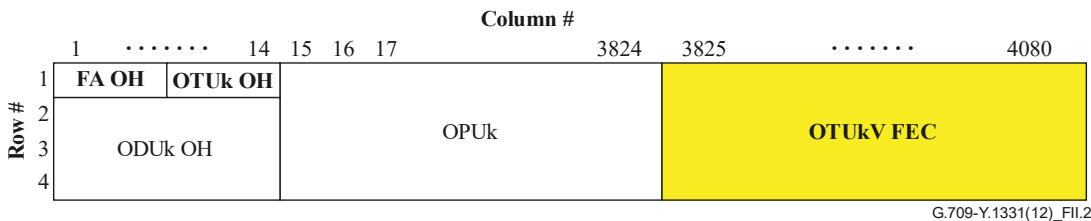


Figure II.2 – OTUk with alternative OTUkV FEC (OTUk-v)

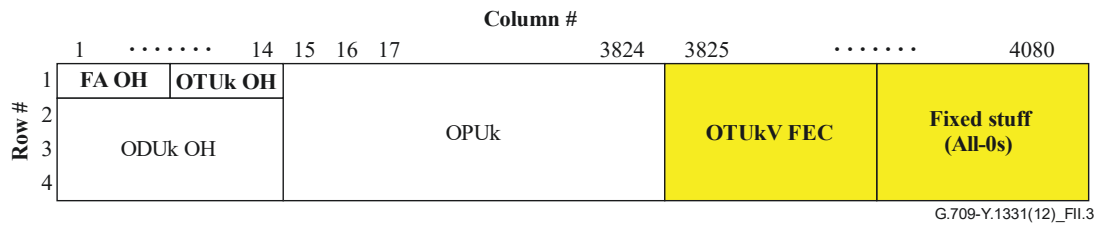


Figure II.3 – OTUk with a smaller OTUkV FEC and the remainder of an FEC area filled with fixed stuff

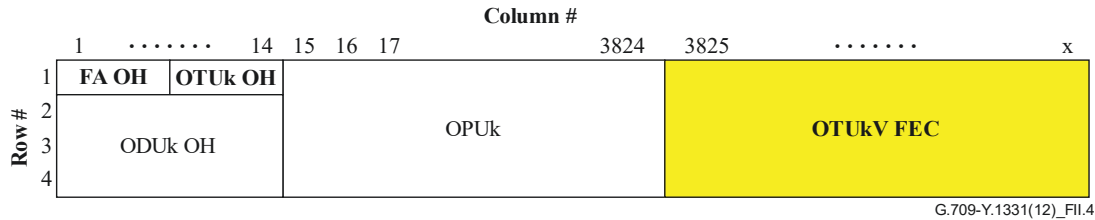


Figure II.4 – OTUk with a larger OTUkV FEC

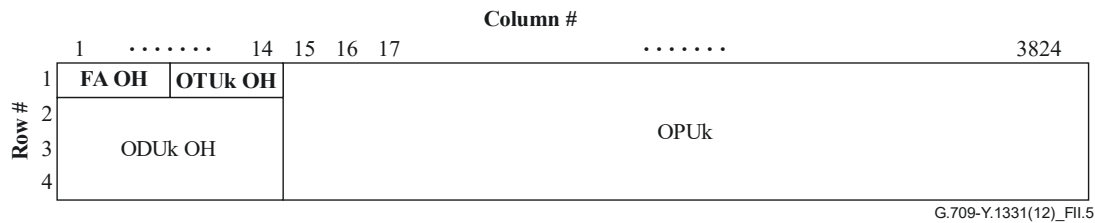


Figure II.5 – OTUk without an OTUkV FEC area

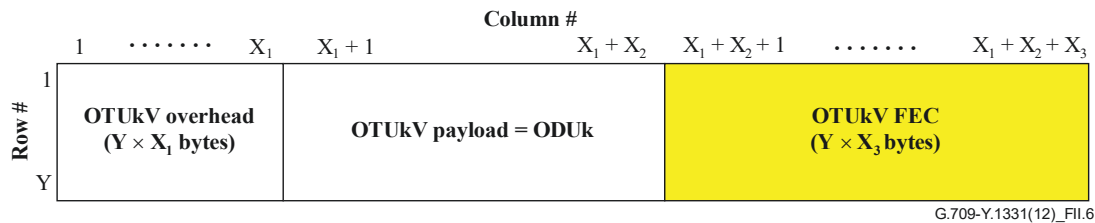


Figure II.6 – OTUkV with a different frame structure

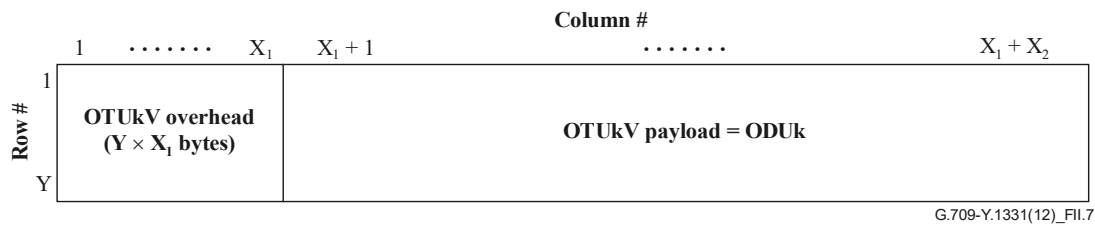


Figure II.7 – OTUkV with a different frame structure and without FEC area

For the case of Figures II.6 and II.7, the mapping of the ODUk signal can be either asynchronous, bit-synchronous, or frame synchronous.

For the case of asynchronous mapping, the ODUk and OTUkV bit rates can be asynchronous. The ODUk signal is mapped as a bit stream into the OTUkV payload area using a stuffing technique.

For the case of bit-synchronous mapping, the ODUk and OTUkV bit rates are synchronous. The ODUk signal is mapped into the OTUkV payload area without stuffing. The ODUk frame is not related to the OTUkV frame.

For the case of a frame synchronous mapping, the ODUk and OTUkV bit rates are synchronous and the frame structures are aligned. The ODUk signal is mapped into the OTUkV payload area without stuffing and with a fixed position of the ODUk frame within the OTUkV frame. (See Figure II.8.)

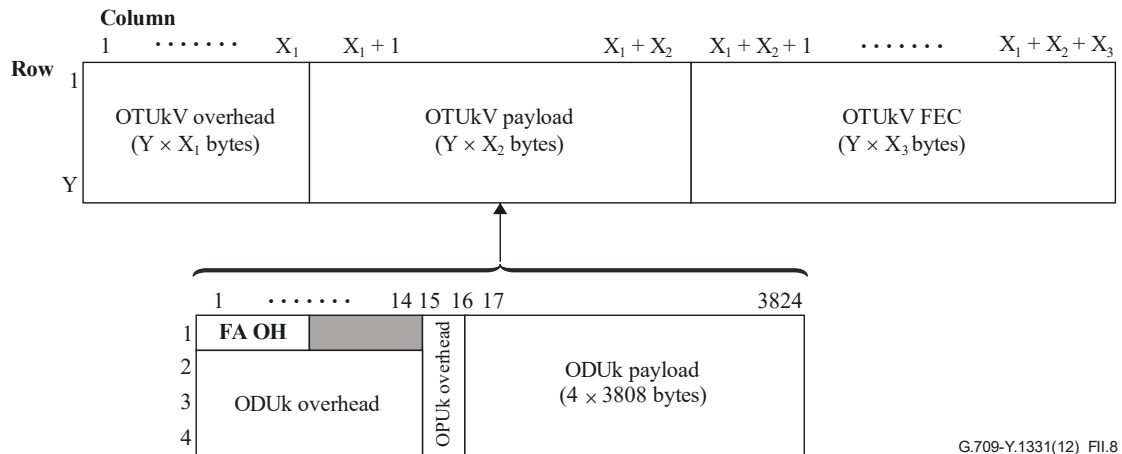


Figure II.8 – Asynchronous (or bit-synchronous) mapping of ODUk into OTUkV

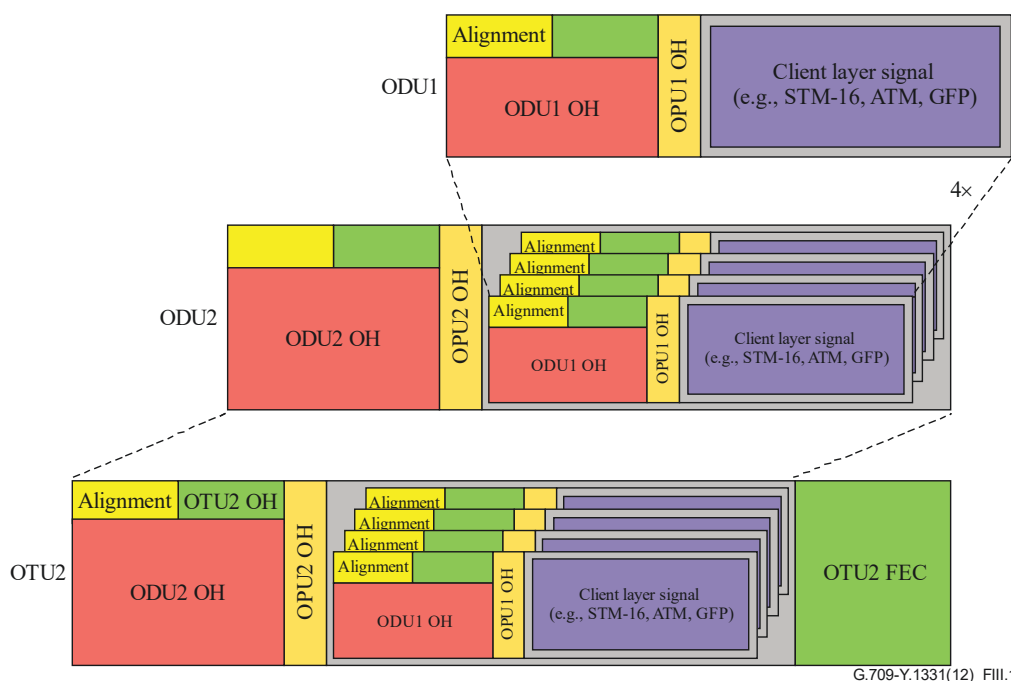
Appendix III

Example of ODUk multiplexing

(This appendix does not form an integral part of this Recommendation.)

Figure III.1 illustrates the multiplexing of four ODU1 signals into an ODU2. The ODU1 signals including the frame alignment overhead and an all-0s pattern in the OTUk overhead locations are adapted to the ODU2 clock via justification (asynchronous mapping). These adapted ODU1 signals are byte interleaved into the OPU2 payload area, and their justification control and opportunity signals (JC, NJO) are frame interleaved into the OPU2 overhead area.

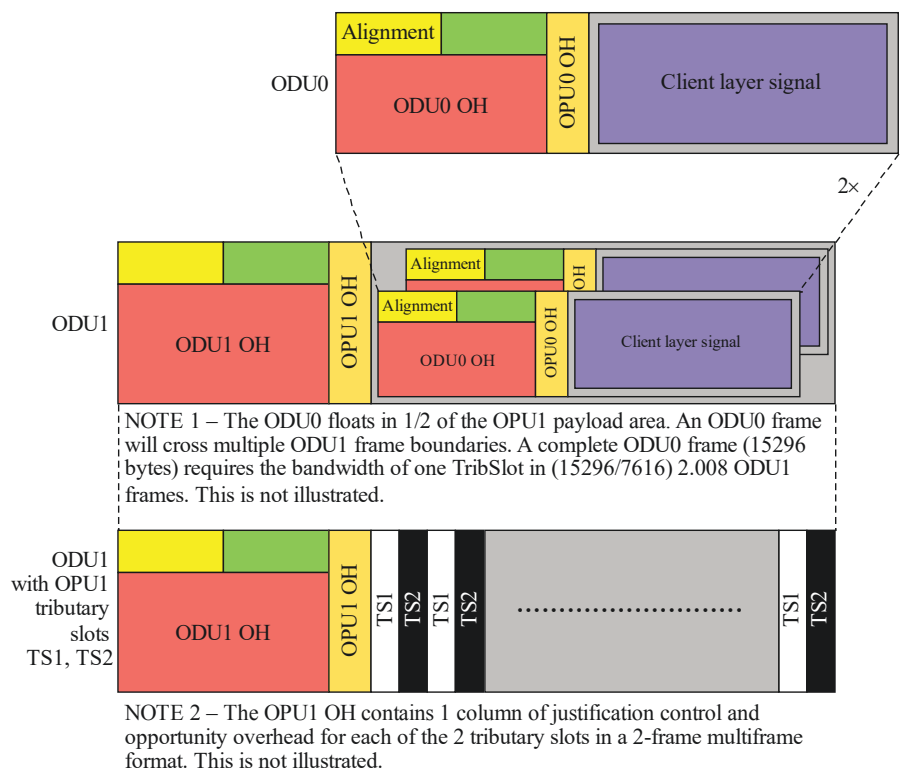
ODU2 overhead is added after which the ODU2 is mapped into the OTU2 [or OTU2V]. OTU2 [or OTU2V] overhead and frame alignment overhead are added to complete the signal for transport via an OTM signal.



NOTE – The ODU1 floats in a quarter of the OPU2 payload area. An ODU1 frame will cross multiple ODU2 frame boundaries. A complete ODU1 frame (15296 bytes) requires the bandwidth of $(15296/3808) 4.017$ ODU2 frames. This is not illustrated.

Figure III.1 – Example of multiplexing 4 ODU1 signals into an ODU2

Figure III.2 illustrates the multiplexing of two ODU0 signals into an ODU1. The ODU0 signals including the frame alignment overhead and an all-0s pattern in the OTUk overhead locations are adapted to the ODU1 clock via justification (asynchronous mapping). These adapted ODU0 signals are byte interleaved into the OPU1 payload area, and their justification control and opportunity signals (JC, NJO) are frame interleaved into the OPU1 overhead area and ODU1 overhead is added.



G.709-Y.1331(12)_FIII.2

Figure III.2 – Example of multiplexing 2 ODU0 signals into an ODU1

Appendix IV

Blank appendix

This appendix is intentionally left blank.

Appendix V

ODUk multiplex structure identifier (MSI) examples

(This appendix does not form an integral part of this Recommendation.)

The following figures present four examples of ODU1 and ODU2 carriage within an OPU3 and the associated MSI encoding.

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	00					000000			<i>TS1</i>
<i>PSI[3]</i>	00					000001			<i>TS2</i>
<i>PSI[4]</i>	00					000010			<i>TS3</i>
<i>PSI[5]</i>	00					000011			<i>TS4</i>
<i>PSI[6]</i>	00					000100			<i>TS5</i>
<i>PSI[7]</i>	00					000101			<i>TS6</i>
<i>PSI[8]</i>	00					000110			<i>TS7</i>
<i>PSI[9]</i>	00					000111			<i>TS8</i>
<i>PSI[10]</i>	00					001000			<i>TS9</i>
<i>PSI[11]</i>	00					001001			<i>TS10</i>
<i>PSI[12]</i>	00					001010			<i>TS11</i>
<i>PSI[13]</i>	00					001011			<i>TS12</i>
<i>PSI[14]</i>	00					001100			<i>TS13</i>
<i>PSI[15]</i>	00					001101			<i>TS14</i>
<i>PSI[16]</i>	00					001110			<i>TS15</i>
<i>PSI[17]</i>	00					001111			<i>TS16</i>

Figure V.1 – OPU3-MSI coding for the case of 16 ODU1s into OPU3

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	01					000000			<i>TS1</i>
<i>PSI[3]</i>	01					000001			<i>TS2</i>
<i>PSI[4]</i>	01					000010			<i>TS3</i>
<i>PSI[5]</i>	01					000011			<i>TS4</i>
<i>PSI[6]</i>	01					000000			<i>TS5</i>
<i>PSI[7]</i>	01					000001			<i>TS6</i>
<i>PSI[8]</i>	01					000010			<i>TS7</i>
<i>PSI[9]</i>	01					000011			<i>TS8</i>
<i>PSI[10]</i>	01					000000			<i>TS9</i>
<i>PSI[11]</i>	01					000001			<i>TS10</i>
<i>PSI[12]</i>	01					000010			<i>TS11</i>
<i>PSI[13]</i>	01					000011			<i>TS12</i>
<i>PSI[14]</i>	01					000000			<i>TS13</i>
<i>PSI[15]</i>	01					000001			<i>TS14</i>
<i>PSI[16]</i>	01					000010			<i>TS15</i>
<i>PSI[17]</i>	01					000011			<i>TS16</i>

**Figure V.2 – OPU3-MSI coding for the case of 4 ODU2s into OPU3 TS#
(1, 5, 9, 13), (2, 6, 10, 14), (3, 7, 11, 15) and (4, 8, 12, 16)**

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	01		000000						TS1
<i>PSI[3]</i>	01		000001						TS2
<i>PSI[4]</i>	01		000001						TS3
<i>PSI[5]</i>	01		000010						TS4
<i>PSI[6]</i>	01		000000						TS5
<i>PSI[7]</i>	01		000011						TS6
<i>PSI[8]</i>	01		000011						TS7
<i>PSI[9]</i>	01		000011						TS8
<i>PSI[10]</i>	01		000000						TS9
<i>PSI[11]</i>	01		000000						TS10
<i>PSI[12]</i>	01		000001						TS11
<i>PSI[13]</i>	01		000001						TS12
<i>PSI[14]</i>	01		000011						TS13
<i>PSI[15]</i>	01		000010						TS14
<i>PSI[16]</i>	01		000010						TS15
<i>PSI[17]</i>	01		000010						TS16

Figure V.3 – OPU3-MSI coding for the case of 4 ODU2s into OPU3 TS# (1, 5, 9, 10), (2, 3, 11, 12), (4, 14, 15, 16) and (6, 7, 8, 13)

	1	2	3	4	5	6	7	8	
<i>PSI[2]</i>	01		000000						TS1
<i>PSI[3]</i>	00		000001						TS2
<i>PSI[4]</i>	00		000010						TS3
<i>PSI[5]</i>	01		000001						TS4
<i>PSI[6]</i>	01		000000						TS5
<i>PSI[7]</i>	00		000101						TS6
<i>PSI[8]</i>	00		000110						TS7
<i>PSI[9]</i>	01		000001						TS8
<i>PSI[10]</i>	01		000000						TS9
<i>PSI[11]</i>	01		000001						TS10
<i>PSI[12]</i>	00		001010						TS11
<i>PSI[13]</i>	00		001011						TS12
<i>PSI[14]</i>	01		000000						TS13
<i>PSI[15]</i>	00		001101						TS14
<i>PSI[16]</i>	00		001110						TS15
<i>PSI[17]</i>	01		000001						TS16

Figure V.4 – OPU3-MSI coding for the case of 5 ODU1s and 2 ODU2s into OPU3 TS# (2), (6), (11), (12), (14), (1, 5, 9, 13) and (4, 8, 10, 16) and OPU3 TS# 3, 7, 15 unallocated (default to ODU1)

Appendix VI

Parallel logic implementation of the CRC-9, CRC-8, CRC-5 and CRC-6

(This appendix does not form an integral part of this Recommendation.)

CRC-9

Table VI.1 illustrates example logic equations for a parallel implementation of the CRC-9 using the $g(x) = x^9 + x^3 + x^2 + 1$ polynomial over the $\Sigma_{C_{nD}}$ fields of JC1, JC2, JC4 and JC5. An "X" in a column of the table indicates that the message bit of that row is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC4.D1 corresponds to bit 2 of the JC4 mapping overhead octet, JC4.D2 corresponds to bit 3 of the JC4 octet, etc. (See Figure 20-7.) After computation, CRC bits crc1 to crc9 are inserted into the JC6 and JC3 octets with crc1 occupying bit 2 of the JC6 octet and crc9 occupying bit 2 of the JC3 octet.

Table VI.1 – Parallel logic equations for the CRC-9 implementation

Mapping overhead bits	CRC checksum bits								
	crc1	crc2	crc3	crc4	crc5	crc6	crc7	crc8	crc9
JC4.D1		X	X			X			
JC4.D2			X	X			X		
JC4.D3				X	X			X	
JC4.D4					X	X			X
JC4.D5	X					X		X	
JC4.D6		X					X		X
JC4.D7	X		X				X		
JC1.D8		X		X				X	
JC1.D9			X		X				X
JC5.D10	X			X		X	X	X	
JC5.D11		X			X		X	X	X
JC5.D12	X		X			X	X		X
JC5.D13	X	X		X					
JC5.D14		X	X		X				
JC5.D15			X	X		X			
JC5.D16				X	X		X		
JC2.D17					X	X		X	
JC2.D18						X	X		X

CRC-8

Table VI.2 illustrates example logic equations for a parallel implementation of the CRC-8 using the $g(x) = x^8 + x^3 + x^2 + 1$ polynomial over the JC1-JC2. An "X" in a column of the table indicates that the message bit of that row is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC1.C1 corresponds to the first bit (MSB) of the first mapping overhead octet (JC1), JC1.C2 corresponds to bit 2 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc8 are inserted into the JC3 octet with crc1 occupying MSB and crc8 the LSB of the octet.

Table VI.2 – Parallel logic equations for the CRC-8 implementation

Mapping overhead bits	CRC checksum bits							
	crc1	crc2	crc3	crc4	crc5	crc6	crc7	crc8
JC1.C1		X				X		X
JC1.C2	X		X			X		
JC1.C3		X		X			X	
JC1.C4			X		X			X
JC1.C5	X			X			X	
JC1.C6		X			X			X
JC1.C7	X		X				X	
JC1.C8		X		X				X
JC2.C9	X		X		X	X	X	
JC2.C10		X		X		X	X	X
JC2.C11	X		X		X	X		X
JC2.C12	X	X		X				
JC2.C13		X	X		X			
JC2.C14			X	X		X		
JC2.II				X	X		X	
JC2.DI					X	X		X

CRC-5

Table VI.3 illustrates example logic equations for a parallel implementation of the CRC-5 using the $g(x) = x^5 + x + 1$ polynomial over the JC4-JC5 C_nD fields. An "X" in a column of the table indicates that the message bit of that row is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC4.D1 corresponds to the first bit (MSB) of the first mapping overhead octet (JC1), JC4.D2 corresponds to bit 2 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc5 are inserted into the JC6 octet with crc1 occupying JC6 bit 4 and crc5 the JC6 bit 8.

Table VI.3 – Parallel logic equations for the CRC-5 implementation

Mapping overhead bits	CRC checksum bits				
	crc1	crc2	crc3	crc4	crc5
JC4.D1	X		X	X	
JC4.D2		X		X	X
JC4.D3	X		X		
JC4.D4		X		X	
JC4.D5			X		X
JC5.D6	X			X	X
JC5.D7	X	X			
JC5.D8		X	X		
JC5.D9			X	X	
JC5.D10				X	X

CRC-6

Table VI.4 illustrates example logic equations for a parallel implementation of the CRC-6 using the $g(x) = x^6 + x^3 + x^2 + 1$ polynomial over the JC1-JC2. An "X" in a column of the table indicates that the message bit of that row is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC1.C1 corresponds to bit 3 of the first mapping overhead octet (JC1), JC1.C2 corresponds to bit 4 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc6 are inserted into the JC3 octet with crc1 occupying bit 3 and crc6 the LSB of the octet. (See Figure 20-7)

Table VI.4 – Parallel logic equations for the CRC-6 implementation

Mapping overhead bits	CRC checksum bits					
	crc1	crc2	crc3	crc4	crc5	crc6
JC1.C1			X		X	X
JC1.C2	X				X	X
JC1.C3	X	X		X	X	X
JC1.C4	X	X	X	X		X
JC1.C5	X	X	X			
JC1.C6		X	X	X		
JC2.C7			X	X	X	
JC2.C8				X	X	X
JC2.C9	X			X		X
JC2.C10	X	X		X		
JC2.II		X	X		X	
JC2.DI			X	X		X

Appendix VII

OTL4.10 structure

(This appendix does not form an integral part of this Recommendation.)

The information of this appendix is included in [b-ITU-T G-Sup.58].

Appendix VIII

CPRI into ODU mapping

(This appendix does not form an integral part of this Recommendation.)

The information of this appendix is included in [b-ITU-T G-Sup.56].

Appendix IX

Overview of CBR clients into OPU mapping types

(This appendix does not form an integral part of this Recommendation.)

As there are many different constant bit rate client signals and multiple mapping procedures, Table IX.1 provides an overview of the mapping procedure that is specified for each client.

Table IX.1 – Overview of CBR client into OPUs mapping types

	OPU0	OPU1	OPU2	OPU2e	OPU3	OPU4	OPUflex
STM-1	GMP with C _{1D}	–	–	–	–	–	–
STM-4	GMP with C _{1D}	–	–	–	–	–	–
STM-16	–	AMP, BMP	–	–	–	–	–
STM-64	–	–	AMP, BMP	–	–	–	–
STM-256	–	–	–	–	AMP, BMP	–	–
1000BASE-X	TTT+GM P no C _{nD}	–	–	–	–	–	–
10GBASE-R	–	–	–	16FS+BMP	–	–	–
40GBASE-R	–	–	–	–	TTT+GMP with C _{8D}	–	–
100GBASE-R	–	–	–	–	–	GMP with C _{8D}	–
FC-100	GMP no C _{nD}	–	–	–	–	–	–
FC-200	–	GMP with C _{8D}	–	–	–	–	–
FC-400	–	–	–	–	–	–	BMP
FC-800	–	–	–	–	–	–	BMP
FC-1200	–	–	–	TTT+16FS+ BMP (Note)	–	–	–
FC-1600	–	–	–	–	–	–	BMP
FC-3200	–	–	–	–	–	–	BMP
CM_GPON	–	AMP	–	–	–	–	–
CM_XGPON	–	–	AMP	–	–	–	–

Table IX.1 – Overview of CBR client into OPUk mapping types

	OPU0	OPU1	OPU2	OPU2e	OPU3	OPU4	OPUflex
IB SDR	–	–	–	–	–	–	BMP
IB DDR	–	–	–	–	–	–	BMP
IB QDR	–	–	–	–	–	–	BMP
SBCON/ESCON	GMP no C _{nD}	–	–	–	–	–	–
DVB_ASI	GMP no C _{nD}	–	–	–	–	–	–
SDI	GMP TBD C _{nD}	–	–	–	–	–	–
1.5G SDI	–	GMP TBD C _{nD}	–	–	–	–	–
3G SDI	–	–	–	–	–	–	BMP
FlexE Client	–	–	–	–	–	–	IMP
FlexE-aware	–	–	–	–	–	–	BGMP
NOTE – For this specific case the mapping used is byte synchronous.							

Appendix X

Overview of ODU_j into OPUs_k mapping types

(This appendix does not form an integral part of this Recommendation.)

As there are many different ODU_j bit rate signals and multiple mapping procedures, Table X.1 provides an overview of the mapping procedure that is specified for each ODU_j.

Table X.1 – Overview of ODU_j client into OPUs_k mapping types

	2.5G tributary slots		1.25G tributary slots			
	OPU2	OPU3	OPU1	OPU2	OPU3	OPU4
ODU0	–	–	ODTU01 AMP (PT=20)	ODTU2.1 GMP (PT=21)	ODTU3.1 GMP (PT=21)	ODTU4.1 GMP (PT=21)
ODU1	ODTU12 AMP (PT=20)	ODTU13 AMP (PT=20)	–	ODTU12 AMP (PT=21)	ODTU13 AMP (PT=21)	ODTU4.2 GMP (PT=21)
ODU2	–	ODTU23 AMP (PT=20)	–	–	ODTU23 AMP (PT=21)	ODTU4.8 GMP (PT=21)
ODU2e	–	–	–	–	ODTU3.9 GMP (PT=21)	ODTU4.8 GMP (PT=21)
ODU3	–	–	–	–	–	ODTU4.31 GMP (PT=21)
ODUflex	–	–	–	ODTU2.ts GMP (PT=21)	ODTU3.ts GMP (PT=21)	ODTU4.ts GMP (PT=21)
ODUflex(IB SDR)	–	–	–	ODTU2.3 GMP (PT=21)	ODTU3.3 GMP (PT=21)	ODTU4.2 GMP (PT=21)
ODUflex(IB DDR)	–	–	–	ODTU2.5 GMP (PT=21)	ODTU3.5 GMP (PT=21)	ODTU4.4 GMP (PT=21)
ODUflex(IB QDR)	–	–	–	–	ODTU3.9 GMP (PT=21)	ODTU4.8 GMP (PT=21)
ODUflex(FC-400)	–	–	–	ODTU2.4 GMP (PT=21)	ODTU3.4 GMP (PT=21)	ODTU4.4 GMP (PT=21)
ODUflex(FC-800)	–	–	–	ODTU2.7 GMP (PT=21)	ODTU3.7 GMP (PT=21)	ODTU4.7 GMP (PT=21)

Table X.1 – Overview of ODU_j client into OPUs mapping types

	2.5G tributary slots		1.25G tributary slots			
	OPU2	OPU3	OPU1	OPU2	OPU3	OPU4
ODUflex(GFP), n=1, ... ,8 (ts=n)	–	–	–	ODTU2.ts (GMP) (PT=21)	ODTU3.ts (GMP) (PT=21)	ODTU4.ts (GMP) (PT=21)
ODUflex(GFP), n=9, ... ,32 (ts=n)	–	–	–	–	ODTU3.ts (GMP) (PT=21)	ODTU4.ts (GMP) (PT=21)
ODUflex(GFP), n=33, ... ,80 (ts=n)	–	–	–	–	–	ODTU4.ts (GMP) (PT=21)

Appendix XI

Derivation of recommended ODUflex(GFP) bit-rates and examples of ODUflex(GFP) clock generation

(This appendix does not form an integral part of this Recommendation.)

XI.1 Introduction

The recommended bit-rates for ODUflex(GFP) are provided in Table 7-8. While in principle an ODUflex(GFP) may be of any bit-rate, there are a variety of reasons for recommending particular rates:

- To encourage a common set of bit-rates which can be expected to be supported by multiple manufacturers.
- To provide the largest amount of bandwidth possible within a given amount of resource (number of tributary slots) independent of the ODUk over which the ODUflex(GFP) may be routed.
- To maintain the number of tributary slots required if the ODUflex(GFP) must be rerouted, e.g., during a restoration.
- To satisfy a protocol requirement for ODUflex hitless resizing that a resizable ODUflex must occupy the same number of tributary slots on every ODUk path over which it is carried, and that a resize operation must always add or remove at least one tributary slot.

XI.2 Tributary slot sizes

ODUflex(GFP) is mapped via GMP into a certain number of 1.25G tributary slots of a HO OPU2, OPU3, or OPU4. Each of these have different tributary slot sizes:

$$OPU2_TS = \frac{238}{237} \times 4 \times STM16 \times \frac{476 \text{ columns}}{3808 \text{ columns}} = 1249409.620 \text{ kbit/s} \pm 20\text{ppm}$$

$$OPU3_TS = \frac{238}{236} \times 16 \times STM16 \times \frac{119 \text{ columns}}{3808 \text{ columns}} = 1254703.729 \text{ kbit/s} \pm 20\text{ppm}$$

$$OPU4_TS = \frac{238}{227} \times 40 \times STM16 \times \frac{47.5 \text{ columns}}{3808 \text{ columns}} = 1301709.251 \text{ kbit/s} \pm 20\text{ppm}$$

An ODUflex(GFP) that occupies 8 or fewer tributary slots may be routed over OPU2, OPU3, or OPU4. The smallest tributary slot that may be encountered along the route of the ODUflex(GFP) is that of OPU2. Even if the initially selected route does not chose a link of OPU2, the ODUflex(GFP) should be sized to a multiple of the OPU2 tributary slot size to preserve the possibility to restore the ODUflex(GFP) over a route that includes OPU2 without changing the size of the ODUflex or the number of tributary slots it occupies.

An ODUflex(GFP) that occupies at least 9, but no more than 32 tributary slots may be routed over OPU3 or OPU4. It does not fit over OPU2. Therefore such an ODUflex may be sized to a multiple of the OPU3 tributary slot size. Even if the initially selected route does not chose a link of OPU3, the ODUflex(GFP) should be sized to a multiple of the OPU3 tributary slot size to preserve the possibility to restore the ODUflex(GFP) over a route that includes OPU3 without changing the size of the ODUflex or the number of tributary slots it occupies.

An ODUflex(GFP) that occupies at least 33, but no more than 80 tributary slots may only be carried via OPU4, and may therefore take advantage of the full size of the OPU4 tributary slot size.

A small margin must be left between the ODUflex(GFP) size and the integral multiple of the tributary slot size to accommodate possible clock variation along a sequence of OPUk links without overflowing the range of C_m in the GMP mapper.

Physical layers for data interfaces such as Ethernet and Fibre Channel have historically used a clock tolerance of ± 100 ppm. This range is sufficiently wide that specifying this as the clock tolerance for ODUflex(GFP) can accommodate a variety of mechanisms for generating an ODUflex(GFP) clock and remain within the clock tolerance range.

ODUk.ts as shown in Table 7-8 is an increment of bandwidth, which, when multiplied by a number of 1.25G tributary slots, gives the recommended size of an ODUflex(GFP) optimized to occupy a given number of tributary slots of a server OPUk. These values are chosen to allow sufficient margin that allows the OPUk and the ODUflex(GFP) to independently vary over their full clock tolerance range without exceeding the capacity of the allocated tributary slots.

The nominal values for ODUk.ts are chosen to be 186 ppm below the bandwidth of a single 1.25G tributary slot of a server OPUk. This allows the ODUflex(GFP) clock to be as much as 100 ppm above its nominal rate and the server OPUk to be as much as 20 ppm below its nominal clock rate, allowing approximately 66 ppm of margin to accommodate jitter and to ensure that the largest average C_m value even in the worst-case situation of the OPUk at -20 ppm from its nominal value and the ODUflex(GFP) at +100 ppm from its nominal value will be one less than the maximum value (i.e., the maximum average C_m is no more than 15231 out of 15232 for ODUflex carried over OPU2 or OPU3, and no more than 15199 out of 15200 for ODUflex carried over OPU4).

Table XI.1 – Generation of ODUflex(GFP) clock from server OPUk clock using fixed C_m

OPUk	Nominal payload bit rate	Nominal 1.25G TS bit rate	ODU2.ts		
			C_m out of	Bit-rate per TS	
OPU2	9'995'276.962	1'249'409.620	15230	1'249'245.570	
			15232		
OPU3	40'150'519.322	1'254'703.729	+20 ppm	1'249'270.555	ODU3.ts
			-20 ppm	1'249'220.585	
			15165 15232	1'249'184.746	Bit-rate per TS
			15230 15232	1'254'538.983	ODU4.ts
OPU4	104'355'975.330	1'301'709.251	+20 ppm	1'249'209.729	C_m out of
			-20 ppm	1'249'159.762	
			14587 15200	1'249'212.687	15198 15200
			+20 ppm	1'249'237.671	1'301'537.974
			-20 ppm	1'249'187.703	1'301'564.004
			nominal	1'249'177.230	1'301'511.943
	ODUk.ts		+100 ppm	1'249'302.148	1'301'467.133
			-100 ppm	1'249'052.312	1'301'597.280
					1'301'336.986

XI.3 Example methods for ODUflex(GFP) clock generation

XI.3.1 Generating ODUflex(GFP) clock from OPUk clock

The clock for an ODUflex(GFP) may be generated from the initial server OPUk over which the ODUflex is carried by setting the value of C_m to a fixed value on the initial segment. Normal GMP processing on subsequent segments avoids the need to couple the server OPUk clocks along the path.

Table XI-1 illustrates how a clock for an ODUflex(GFP) occupying $n \times \text{ODUk.ts}$ can be derived from the server OPUk clock using a fixed value of C_m in the initial segment of the path.

For example, for an ODUflex(GFP) occupying up to 8 tributary slots should be based on ODU2.ts, and therefore have a clock frequency of $n \times 1'249'177.230 \text{ kbit/s} \pm 100 \text{ ppm}$. This allows the ODUflex(GFP) to have a frequency of between $n \times 1'249'052.312 \text{ kbit/s}$ and $n \times 1'249'302.148 \text{ kbit/s}$.

- If the initial segment over which the ODUflex(GFP) is carried is an OPU2, a clock in this range can be generated by fixing the value of C_m on the initial segment to 15230, which will result in the ODUflex having a clock of $n \times 1'249'245.570 \text{ kbit/s} \pm 20 \text{ ppm}$. While the center frequency of this range differs from the nominal value of ODU2.ts, the clock tolerance is narrower, being locked to the OPU2, so the possible clock range is fully within the $\pm 100 \text{ ppm}$ range allowed.
- If the initial segment is an OPU3, the ODUflex(GFP) of a multiple of ODU2.ts can be generated using a fixed value of $C_m=15165$ on the initial ODU3 segment, which will result in the ODUflex having a clock of $n \times 1'249'184.746 \text{ kbit/s} \pm 20 \text{ ppm}$,
- If the initial segment is an OPU4, the ODUflex(GFP) of a multiple of ODU2.ts can be generated using a fixed value of $C_m=14587$ on the initial OPU4 segment, which will result in the ODUflex having a clock of $n \times 1'249'212.687 \text{ kbit/s} \pm 20 \text{ ppm}$.

The center frequencies of all of these ODUflex(GFP) are slightly different, but the resulting ranges for the clocks all fall within the $\pm 100 \text{ ppm}$ window (see Figure XI.1). Fixed C_m for generating ODU3.ts and ODU4.ts from the initial OPUk can similarly be found from this table.

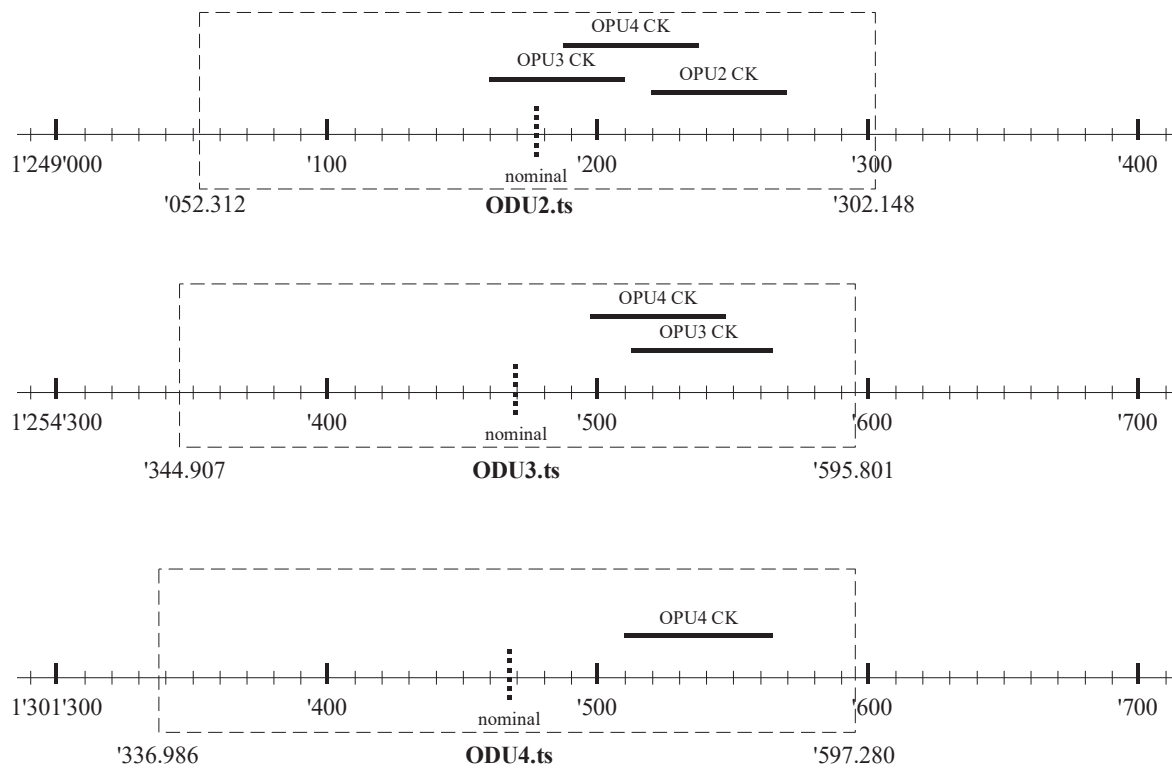


Figure XI.1 – Graphical representation of frequency ranges in Table XI.1

XI.3.2 Generating ODUflex(GFP) clock from system clock

The clock for an ODUflex(GFP) may be generated using a multiplier from the internal system clock. Normally the internal system clock will have an accuracy of at least ± 20 ppm, perhaps even ± 4.6 ppm for a network element that supports both SDH and OTN interfaces. The exact multiplier to be used is implementation specific, and be chosen so that the range of the generated clock falls within the specified ± 100 ppm window around the nominal value of $n \times \text{ODUk.ts}$.

Appendix XII

Terminology changes between ITU-T G.709 Edition 4 and Edition 5

(This appendix does not form an integral part of this Recommendation.)

In Edition 5 of this Recommendation a number of terms used in Editions 1 to 4 have been modified. Table XII.1 provides the mapping between these terms in Editions 1 to 4 and Edition 5.

Table XII.1 – Terminology mapping

ITU-T G.709 Edition 1 to 4	ITU-T G.709 Edition 5
OTM-0.m	SOTU
OTM-0.mvn	(multi-lane) SOTU
OTM-nr.m	MOTU
OTM-l.m	SOTUm
OTM-n.m	MOTUm
Optical Channel (OCh)	OCh
Optical Channel with reduced functionality (OChr)	OCh-P
Optical Channel Payload (OCh-P)	OCh-P
Optical Channel Overhead (OCh-OH)	OCh-O
Optical channel Transport Unit (OTU)	Optical Transport Unit (OTU)
Optical channel Data Unit (ODU)	Optical Data Unit (ODU)
Optical channel Payload Unit (OPU)	Optical Payload Unit (OPU)
Optical channel Data Tributary Unit (ODTU)	Optical Data Tributary Unit (ODTU)
Optical channel Data Tributary Unit Group (ODTUG)	Optical Data Tributary Unit Group (ODTUG)
Optical Multiplex Unit (OMU)	-
Optical Channel Carrier (OCC)	-
Optical Channel Carrier - overhead (OCCo)	-
Optical Channel Carrier – payload (OCCp)	-
Optical Channel Carrier with reduced functionality (OCCr)	-
Optical Carrier Group (OCG)	-
Optical Carrier Group with reduced functionality (OCGr)	-
Optical Carrier Group of order n (OCG)	-
OTM Overhead Signal (OOS)	Optical Supervisory Channel (OSC)
Optical Transport Lane (OTLk.n)	OTLk.n lane
Optical Transport Lane Carrier Group (OTLCG)	-
Optical Transport Lane Carrier (OTLC)	-
OPS0	OPS
OPSn	OPS
OPSMnk	OPS
OTS-OH	OTS-O
OMS-OH	OMS-O

In Edition 5.0 of this Recommendation the following new terms are introduced:

Table XII.2 – New terms

Optical Tributary Signal (OTSi)
Optical Tributary Signal Group (OTSiG)
Optical Tributary Signal Group - Overhead (OTSiG-O)
Optical Tributary Signal Assembly (OTSiA)

The relationship between these new terms in and the OCh, OChr, OPS0, OPSn and OPSM terms is as follows:

Table XII.3 – Relationship between OCh, OChr, OPS0, OPSn, OPSM and OTSi terminology

OCh-P	OTSiG with one OTSi
OCh-O	Reduced version of OTSiG-O: - FDI-P, FDI-O and OCI: supported - TTI, BDI-P, BDI-O, TSI: not supported
OCh	Reduced version of OTSiA: - OTSiG with one or more OTSi - FDI-P, FDI-O and OCI: supported - TTI, BDI-P, BDI-O, TSI: not supported
OChr	OTSiG with one OTSi; no OTSiG-O
OPS0	OTSiG with one OTSi; no OTSiG-O
OPSMnk	OTSiG with m OTSi; no OTSiG-O
OPSn	n OTSiG and no OTSiG-O; each OTSiG with one OTSi

Appendix XIII

OTUCn sub rates (OTUCn-M) Applications

(This appendix does not form an integral part of this Recommendation.)

XIII.1 Introduction

This appendix describes OTUCn-M application scenarios for the OTUCn-M signal defined in Annex H. It is assumed that the OTUCn-M processing is performed in the PHY module so that the interface between the module and framer (MFI) is not changed.

Figures XIII.1 and XIII.2 illustrate two scenarios for subrating applications. The first scenario deploys subrating between two line ports connecting two L1/L0 ODU cross connects (XC). The second scenario deploys subrating between transponders which are in a different domain B, which are separated from the L1 ODU XCs in domain A and/or C.

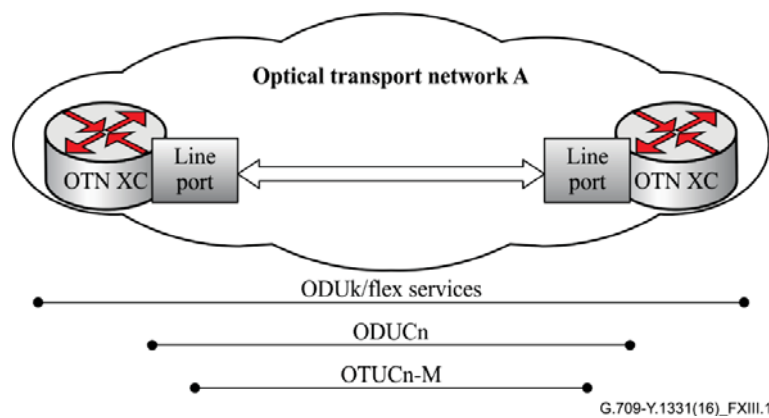


Figure XIII.1 – OTUCn sub rate application scenario A

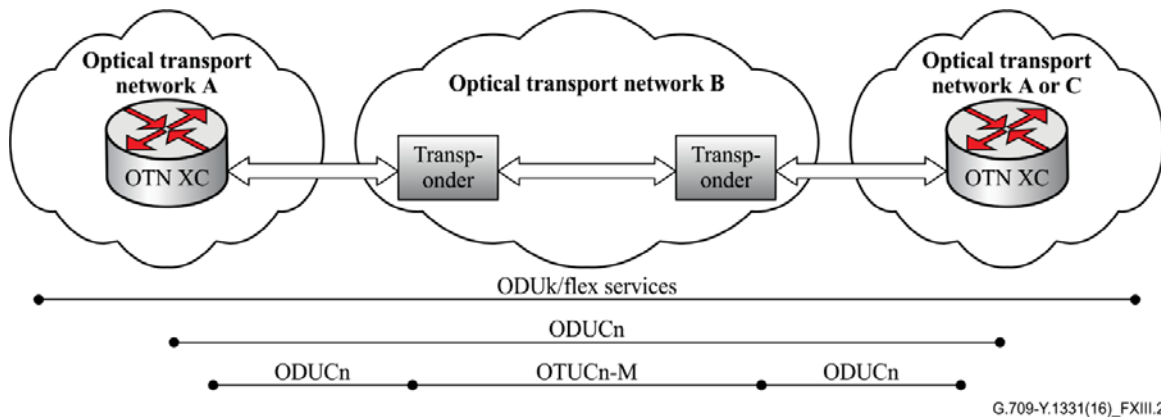


Figure XIII.2 – OTUCn sub rate application scenario B

XIII.2 OTUCn-M frame format and rates

As defined in Annex H, the OTUCn-M frame is a type of OTUCn frame which contains n instances of OTUC, ODUC and OPUC overhead and M 5 Gbit/s OPUCn tributary slots. If a particular value of M is not indicated, the frame contains 20*n tributary slots.

The use of this notation is illustrated below:

- OTUC2-25: a 125 Gbit/s OTU signal with 25 (5 Gbit/s) tributary slots and 2 instances of OxUC overhead

- OTUC2-30: a 150 Gbit/s OTU signal with 30 (5 Gbit/s) tributary slots and 2 instances of O_xUC overhead
- OTUC2: a 200 Gbit/s OTU signal with 40 (5 Gbit/s) tributary slots and 2 instances of O_xUC overhead
- OTUC3-50: a 250 Gbit/s OTU signal with 50 (5 Gbit/s) tributary slots and 3 instances of O_xUC overhead

The bit rate of an OTUC_n-M signal is $(10n + 119 * M) / (10n + 119 * 20 * n) * \text{OTUC}_n \text{ bit rate}$. . The choice of which TS are assigned to each of the OTUC instances, and hence the signal rate of that OTUC instance, is vendor or application specific.

XIII.3 OTUC_n-M fault condition

For the case where subrating is used between transponders the following checks should be made.

In the transmit (source) direction, transponder X should verify that the OPU tributary slots with the Availability bit of their OPUC_n MSI overhead set to "available" are carried transparently by the OTUC_n-M. For the case where one or more availability bits for the the tributary slots that are not carried transparently are set to "available", an OTUC_n-AIS should be inserted by Transponder X.

In the receive (sink) direction, transponder Y should check for a mismatch between received and expected OTUC_n-M. If such a mismatch is detected an OTUC_n-AIS should be inserted by Transponder Y.

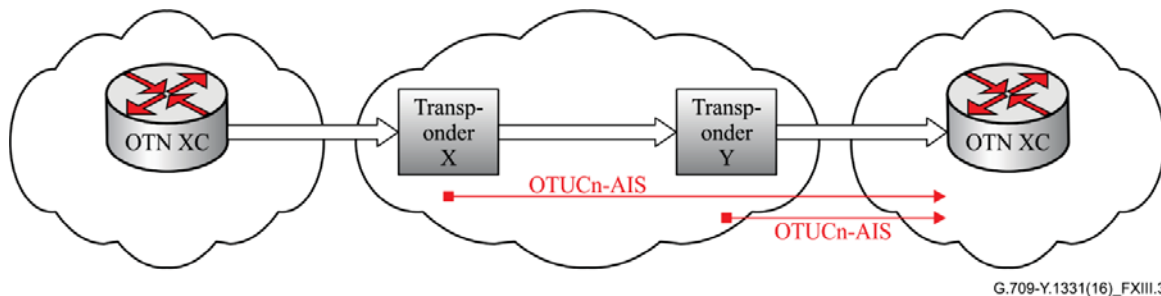


Figure XIII.3 – OTUC_n sub rate mismatch fault condition

Bibliography

- [b-ITU-T G-Sup.43] ITU-T G-series Recommendations – Supplement 43 (2011), *Transport of IEEE 10GBASE-R in optical transport networks (OTN)*.
- [b-ITU-T G-Sup.56] ITU-T G-series Recommendations – Supplement 56 (2016), *OTN Transport of CPRI signals*.
- [b-ITU-T G-Sup.58] ITU-T G-series Recommendations – Supplement 58 (2016), *Optical transport network (OTN) module framer interfaces (MFIs)*.
- [b-IEEE 1588] IEEE 1588:2013, Precision Time Protocol Time Synchronization Performance.
- [b-ANSI INCITS 296] ANSI INCITS 296-1997, *Information Technology – Single-Byte Command Code Sets CONnection (SBCON) Architecture*.
- [b-ANSI INCITS 364] ANSI INCITS 364-2003, *Information Technology – Fibre Channel 10 Gigabit (10GFC)*.
- [b-INCITS 488] INCITS 488:2016, *Information Technology – Fibre Channel Framing and Signaling – 4 (FC-FS-4)*.
- [b-IB ARCH] InfiniBand Trade Association (2006), *InfiniBand Architecture Specification, Volume 2, Release 1.2.1*.
- [b-INCITS 470] INCITS 470:2011, *Information Technology – Fibre Channel – Framing and Signaling – 3 (FC-FS-3)*.
- [b-CPRI] CPRI Specification V5.0 (2011-09-21) Common Public Radio Interface (CPRI), *Interface Specification*.
- [b-IETF RFC 6205] IETF RFC 6205 (2011), Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers.

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems